

COMMITTEE DRAFT		Reference document: SC 27 N18705	
ISO/IEC CD 18045 (revision)			
Date: 2018-07-05		Supersedes document WG 3 N1478	
THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.			
ISO/IEC JTC 1/SC 27 Information technology - Security techniques Secretariat: Germany (DIN)		Circulated to P- and O-members, and to technical committees and organizations in liaison for comments by: 2018-08-30 Please submit your comments via the online balloting application by the due date indicated.	
ISO/IEC 1 st CD 18045 (revision)			
Title: IT Security techniques — Evaluation criteria for IT security — Methodology for IT security evaluation			
Project: ISO/IEC 18045 (revision)			
Explanatory Report			
Status	SC 27 Decision	Reference documents	
		Input	Output
Study Period IT security testing, evaluation and assurance standards and techniques	51 st WG 3 meeting, Oct. 2015, Recommendations 5, 6 (N15594 = WG 3 N1251).		Terms of Reference (WG 5 N1258); 1 st /2 nd call f. contr. (WG 3 N1259 /1317)..
	52 nd WG 3 meeting, April 2016, Recommendation 5, 7 (N16026 = WG 3 N1296).	Expert contr. (WG 3 N1299, 1301).	3 rd call f. contr. (WG 3 N1377); Rapporteur's report (WG 3 N1320); Liaison to: PRIPARE (WG 5 N = N16266).
ISO/IEC NP 18045	53 rd WG 3 meeting, Oct. 2016, Recommendations 6, 15 (N16800 = WG 5 N600).	Expert contr. (WG 3 N1368, N1371, N1373).	SP report (WG 3 N1363); Call f. editor (WG 3 N1387 = N16886); Liaisons to: CCDB (WG 3 N1330); The Open Group (WG 3 N1332); Text f. NWIP (N16884).
ISO/IEC NP 18045 1 st WD	54th WG 3 meeting, April 2017, Recommendations 5,10 11, 14 (N17041 = WG 3 N1413).	SoV (N17030).	Liaisons to: CCDB (WG 3 N1391); The Open Group (WG 3 N1394); ISO/TC 22/SC 32 (N17373); Text f. 1st WD (WG 3 N1440).
ISO/IEC 18045 2 nd WD	55th WG 3 meeting, October / November 2017, Recommendations 8, 10, 15 (N17666 = WG 3 N1494).	SoCom (WG 3 N1476); Draft DoC (WG 3 N1501).	Liaison to ISO/TC 22/SC 32/WG 11 (N18103); Status (WG 3 N1465); DoC (WG 3 N1462); Text f. 2 nd WD (WG 3 N1478).
ISO/IEC 18045 1 st CD	56th WG 3 meeting, April 2018, Recommendations 10, 12 / 30 th SC 27 Plenary, April 2018, Resolution 6 (N18710) (N18471 = WG 3 N1557).	SoCom (WG 3 N1536); Late Com (WG 3 N1567); Draft DoC (WG 3 N15).	DoC (WG 3 N1527); Text f. 1 st CD (N18705).
CD Registration and Consideration			
In accordance with resolution 6 (see SC 27 N18710) of the 30 th SC 27 Plenary meeting held in Wuhan, China, 2018-04-23/24 the hereby attached document has been registered with the ISO Central Secretariat (ITTF) as 1st Committee			
Draft (CD) and is being circulated for a 1 st CD 8 weeks letter ballot closing by 2018-08-30			
Medium: http://isotc.iso.org/livelink/livelink/open/jtc1sc27			
No. of pages: 1+ 502			

1
2
3
4

ISO/IEC 18045:####(EN)

ISO/IEC JTC 1/SC 27/WG 3 [N18705](#)

Secretariat: DIN

5 **IT security techniques — Evaluation criteria for IT security — Methodology**
6 **for IT security evaluation**

7

8 **CD stage**

9

10
11
12
13
14

Warning for WDs and CDs
This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.
Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
copyright@iso.org
www.iso.org

28	Contents	Page
29	1	Scope 1
30	2	Normative references 1
31	3	Terms and definitions 1
32	4	Symbols and abbreviated terms 1
33	5	Overview 1
34	5.1	Organisation of this International Standard 1
35	6	Document Conventions 2
36	6.1	Terminology 2
37	6.2	Verb usage 2
38	6.3	General evaluation guidance 2
39	6.4	Relationship between ISO/IEC 15408 and ISO/IEC 18045 structures 2
40	7	Evaluation process and related tasks 3
41	7.1	Introduction 3
42	7.2	Evaluation process overview 4
43	7.2.1	Objectives 4
44	7.2.2	Responsibilities of the roles 4
45	7.2.3	Relationship of roles 4
46	7.2.4	General evaluation model 4
47	7.2.5	Evaluator verdicts 5
48	7.3	Evaluation input task 7
49	7.3.1	Objectives 7
50	7.3.2	Application notes 7
51	7.3.3	Management of evaluation evidence sub-task 8
52	7.4	Evaluation sub-activities 8
53	7.5	Evaluation output task 8
54	7.5.1	Objectives 8
55	7.5.2	Management of evaluation outputs 9
56	7.5.3	Application notes 9
57	7.5.4	Write OR sub-task 9
58	7.5.5	Write ETR sub-task 9
59	8	Class APE: Protection Profile evaluation 15
60	8.1	Introduction 15
61	8.2	Application notes 15
62	8.2.1	Re-using the evaluation results of certified PPs 15
63	8.3	PP introduction (APE_INT) 16
64	8.3.1	Evaluation of sub-activity (APE_INT.1) 16
65	8.4	Conformance claims (APE_CCL) 17
66	8.4.1	Evaluation of sub-activity (APE_CCL.1) 17
67	8.5	Security problem definition (APE_SPD) 26
68	8.5.1	Evaluation of sub-activity (APE_SPD.1) 26
69	8.6	Security objectives (APE_OBJ) 27
70	8.6.1	Evaluation of sub-activity (APE_OBJ.1) 27
71	8.6.2	Evaluation of sub-activity (APE_OBJ.2) 28
72	8.7	Extended components definition (APE_ECD) 30
73	8.7.1	Evaluation of sub-activity (APE_ECD.1) 30
74	8.8	Security requirements (APE_REQ) 34
75	8.8.1	Evaluation of sub-activity (APE_REQ.1) 34
76	8.8.2	Evaluation of sub-activity (APE_REQ.2) 41
77	9	Class ACE: Protection Profile Configuration evaluation 45

78	9.1	Introduction	45
79	9.2	PP-Module introduction (ACE_INT)	46
80	9.2.1	Evaluation of sub-activity (ACE_INT.1)	46
81	9.3	PP-Module conformance claims (ACE_CCL)	47
82	9.3.1	Evaluation of sub-activity (ACE_CCL.1)	47
83	9.4	PP-Module Security problem definition (ACE_SPD)	49
84	9.4.1	Evaluation of sub-activity (ACE_SPD.1)	49
85	9.5	PP-Module Security objectives (ACE_OBJ)	50
86	9.5.1	Evaluation of sub-activity (ACE_OBJ.1)	50
87	9.6	PP-Module extended components definition (ACE_ECD)	50
88	9.6.1	Evaluation of sub-activity (ACE_ECD.1)	50
89	9.7	PP-Module security requirements (ACE_REQ)	50
90	9.7.1	Evaluation of sub-activity (ACE_REQ.1)	50
91	9.8	PP-Module consistency (ACE_MCO)	50
92	9.8.1	Evaluation of sub-activity (ACE_MCO.1)	50
93	9.9	PP-Configuration consistency (ACE_CCO)	52
94	9.9.1	Evaluation of sub-activity (ACE_CCO.1)	52
95	10	Class ASE: Security Target evaluation	55
96	10.1	Introduction	55
97	10.2	Application notes	55
98	10.2.1	Re-using the evaluation results of certified PPs	55
99	10.3	ST introduction (ASE_INT)	56
100	10.3.1	Evaluation of sub-activity (ASE_INT.1)	56
101	10.4	Conformance claims (ASE_CCL)	59
102	10.4.1	Evaluation of sub-activity (ASE_CCL.1)	59
103	10.5	Security problem definition (ASE_SPD)	71
104	10.5.1	Evaluation of sub-activity (ASE_SPD.1)	71
105	10.6	Security objectives (ASE_OBJ)	72
106	10.6.1	Evaluation of sub-activity (ASE_OBJ.1)	72
107	10.6.2	Evaluation of sub-activity (ASE_OBJ.2)	73
108	10.7	Extended components definition (ASE_ECD)	75
109	10.7.1	Evaluation of sub-activity (ASE_ECD.1)	75
110	10.8	Security requirements (ASE_REQ)	79
111	10.8.1	Evaluation of sub-activity (ASE_REQ.1)	79
112	10.8.2	Evaluation of sub-activity (ASE_REQ.2)	85
113	10.9	TOE summary specification (ASE_TSS)	90
114	10.9.1	Evaluation of sub-activity (ASE_TSS.1)	90
115	10.9.2	Evaluation of sub-activity (ASE_TSS.2)	90
116	10.10	[PLACE-HOLDER] ST Additional Module Analysis (ASE_AMA)	92
117	11	Class ADV: Development	92
118	11.1	Introduction	92
119	11.2	Application notes	92
120	11.3	Security Architecture (ADV_ARC)	93
121	11.3.1	Evaluation of sub-activity (ADV_ARC.1)	93
122	11.4	Functional specification (ADV_FSP)	97
123	11.4.1	Evaluation of sub-activity (ADV_FSP.1)	97
124	11.4.2	Evaluation of sub-activity (ADV_FSP.2)	101
125	11.4.3	Evaluation of sub-activity (ADV_FSP.3)	105
126	11.4.4	Evaluation of sub-activity (ADV_FSP.4)	111
127	11.4.5	Evaluation of sub-activity (ADV_FSP.5)	116
128	11.4.6	Evaluation of sub-activity (ADV_FSP.6)	122
129	11.5	Implementation representation (ADV_IMP)	122
130	11.5.1	Evaluation of sub-activity (ADV_IMP.1)	122
131	11.5.2	Evaluation of sub-activity (ADV_IMP.2)	124
132	11.6	TSF internals (ADV_INT)	127
133	11.6.1	Evaluation of sub-activity (ADV_INT.1)	127
134	11.6.2	Evaluation of sub-activity (ADV_INT.2)	130
135	11.6.3	Evaluation of sub-activity (ADV_INT.3)	132
136	11.7	[PLACE-HOLDER] TOE Modular Traceability of Functional Requirements in Code (ADV_MTC)	135

137	11.8	Suggestions for text would be welcomed in response to CD1 review. If none are received then this topic will be left to the next revision. Security policy modelling (ADV_SPM).....	135
138			
139	11.8.1	Evaluation of sub-activity (ADV_SPM.1)	135
140	11.9	TOE design (ADV_TDS)	140
141	11.9.1	Evaluation of sub-activity (ADV_TDS.1)	140
142	11.9.2	Evaluation of sub-activity (ADV_TDS.2)	143
143	11.9.3	Evaluation of sub-activity (ADV_TDS.3)	148
144	11.9.4	Evaluation of sub-activity (ADV_TDS.4)	158
145	11.9.5	Evaluation of sub-activity (ADV_TDS.5)	167
146	11.9.6	Evaluation of sub-activity (ADV_TDS.6)	175
147	12	Class AGD: Guidance documents.....	175
148	12.1	Introduction	175
149	12.2	Application notes	175
150	12.3	Operational user guidance (AGD_OPE)	176
151	12.3.1	Evaluation of sub-activity (AGD_OPE.1).....	176
152	12.4	Preparative procedures (AGD_PRE).....	179
153	12.4.1	Evaluation of sub-activity (AGD_PRE.1).....	179
154	13	Class ALC: Life-cycle support	180
155	13.1	Introduction	180
156	13.2	CM capabilities (ALC_CMC).....	181
157	13.2.1	Evaluation of sub-activity (ALC_CMC.1)	181
158	13.2.2	Evaluation of sub-activity (ALC_CMC.2)	182
159	13.2.3	Evaluation of sub-activity (ALC_CMC.3)	184
160	13.2.4	Evaluation of sub-activity (ALC_CMC.4)	188
161	13.2.5	Evaluation of sub-activity (ALC_CMC.5)	193
162	13.3	CM scope (ALC_CMS).....	200
163	13.3.1	Evaluation of sub-activity (ALC_CMS.1)	200
164	13.3.2	Evaluation of sub-activity (ALC_CMS.2)	201
165	13.3.3	Evaluation of sub-activity (ALC_CMS.3)	202
166	13.3.4	Evaluation of sub-activity (ALC_CMS.4)	203
167	13.3.5	Evaluation of sub-activity (ALC_CMS.5)	204
168	13.4	Delivery (ALC_DEL).....	205
169	13.4.1	Evaluation of sub-activity (ALC_DEL.1)	205
170	13.5	Development security (ALC_DVS)	207
171	13.5.1	Evaluation of sub-activity (ALC_DVS.1)	207
172	13.5.2	Evaluation of sub-activity (ALC_DVS.2)	209
173	13.6	Flaw remediation (ALC_FLR)	213
174	13.6.1	Evaluation of sub-activity (ALC_FLR.1)	213
175	13.6.2	Evaluation of sub-activity (ALC_FLR.2)	215
176	13.6.3	Evaluation of sub-activity (ALC_FLR.3)	218
177	13.7	Life-cycle definition (ALC_LCD).....	223
178	13.7.1	Evaluation of sub-activity (ALC_LCD.1)	223
179	13.7.2	Evaluation of sub-activity (ALC_LCD.2)	224
180	13.8	Tools and techniques (ALC_TAT)	226
181	13.8.1	Evaluation of sub-activity (ALC_TAT.1).....	226
182	13.8.2	Evaluation of sub-activity (ALC_TAT.2).....	228
183	13.8.3	Evaluation of sub-activity (ALC_TAT.3).....	231
184	14	Class ASE: Security Target evaluation	233
185	14.1	[PLACE-HOLDER] ST Additional Module Analysis (ASE_AMA)	233
186	15	Suggestions for text would be welcomed in response to CD1 review. If none are received then this topic will be left to the next revision. Class ATE: Tests	234
187			
188	15.1	Introduction	234
189	15.2	Application notes	234
190	15.2.1	Understanding the expected behaviour of the TOE.....	234
191	15.2.2	Testing vs. alternate approaches to verify the expected behaviour of functionality.....	235
192	15.2.3	Verifying the adequacy of tests	235
193	15.3	Coverage (ATE_COV)	236
194	15.3.1	Evaluation of sub-activity (ATE_COV.1)	236

195	15.3.2	Evaluation of sub-activity (ATE_COV.2).....	236
196	15.3.3	Evaluation of sub-activity (ATE_COV.3).....	238
197	15.4	Depth (ATE_DPT).....	240
198	15.4.1	Evaluation of sub-activity (ATE_DPT.1).....	240
199	15.4.2	Evaluation of sub-activity (ATE_DPT.2).....	242
200	15.4.3	Evaluation of sub-activity (ATE_DPT.3).....	245
201	15.4.4	Evaluation of sub-activity (ATE_DPT.4).....	247
202	15.5	Functional tests (ATE_FUN).....	248
203	15.5.1	Evaluation of sub-activity (ATE_FUN.1).....	248
204	15.5.2	Evaluation of sub-activity (ATE_FUN.2).....	251
205	15.6	Independent testing (ATE_IND).....	254
206	15.6.1	Evaluation of sub-activity (ATE_IND.1).....	254
207	15.6.2	Evaluation of sub-activity (ATE_IND.2).....	258
208	15.6.3	Evaluation of sub-activity (ATE_IND.3).....	264
209	15.7	[PLACE-HOLDER] TOE Modular Testing Knowledge (ATE_MTK).....	264
210	15.8	Suggestions for text would be welcomed in response to CD1 review. If none are received then this	
211		topic will be left to the next revision. [PLACE-HOLDER] TOE Modular Traceability of	
212		Functional Requirements in Tests (ATE_MTT).....	264
213	16	Class AVA: Vulnerability assessment.....	264
214	16.1	Introduction.....	264
215	16.2	Vulnerability analysis (AVA_VAN).....	264
216	16.2.1	Evaluation of sub-activity (AVA_VAN.1).....	264
217	16.2.2	Evaluation of sub-activity (AVA_VAN.2).....	270
218	16.2.3	Evaluation of sub-activity (AVA_VAN.3).....	276
219	16.2.4	Evaluation of sub-activity (AVA_VAN.4).....	285
220	16.2.5	Evaluation of sub-activity (AVA_VAN.5).....	292
221	17	Class ACO: Composition.....	300
222	17.1	Introduction.....	300
223	17.2	Application notes.....	301
224	17.3	Composition rationale (ACO_COR).....	302
225	17.3.1	Evaluation of sub-activity (ACO_COR.1).....	302
226	17.4	Development evidence (ACO_DEV).....	308
227	17.4.1	Evaluation of sub-activity (ACO_DEV.1).....	308
228	17.4.2	Evaluation of sub-activity (ACO_DEV.2).....	309
229	17.4.3	Evaluation of sub-activity (ACO_DEV.3).....	311
230	17.5	Reliance of dependent component (ACO_REL).....	313
231	17.5.1	Evaluation of sub-activity (ACO_REL.1).....	313
232	17.5.2	Evaluation of sub-activity (ACO_REL.2).....	315
233	17.6	Composed TOE testing (ACO_CTT).....	318
234	17.6.1	Evaluation of sub-activity (ACO_CTT.1).....	318
235	17.6.2	Evaluation of sub-activity (ACO_CTT.2).....	321
236	17.7	Composition vulnerability analysis (ACO_VUL).....	324
237	17.7.1	Evaluation of sub-activity (ACO_VUL.1).....	324
238	17.7.2	Evaluation of sub-activity (ACO_VUL.2).....	327
239	17.7.3	Evaluation of sub-activity (ACO_VUL.3).....	331
240	Annex A	(informative) General evaluation guidance.....	335
241	A.1	Objectives.....	335
242	A.2	Sampling.....	335
243	A.3	Dependencies.....	337
244	A.3.1	Dependencies between activities.....	337
245	A.3.2	Dependencies between sub-activities.....	337
246	A.3.3	Dependencies between actions.....	337
247	A.4	Site Visits.....	338
248	A.4.1	Introduction.....	338
249	A.4.2	General Approach.....	338
250	A.4.3	Orientation Guide for the Preparation of the Check List.....	339
251	A.4.4	Example of a checklist.....	341
252	A.5	Scheme Responsibilities.....	343

253	Annex B (informative) Vulnerability Assessment (AVA)	345
254	B.1 What is Vulnerability Analysis	345
255	B.2 Evaluator construction of a Vulnerability Analysis	345
256	B.2.1 Generic vulnerability guidance	346
257	B.2.2 Identification of Potential Vulnerabilities	353
258	B.3 When attack potential is used	357
259	B.3.1 Developer	357
260	B.3.2 Evaluator	357
261	B.4 Calculating attack potential	358
262	B.4.1 Application of attack potential	358
263	B.4.2 Characterising attack potential	359
264	B.5 Example calculation for direct attack	365
265	Annex C Evaluation Techniques and Tools (informative)	367
266	C.1 Semiformal and formal methods	367
267	C.1.1 Description of styles	367
268	C.1.2 Security policy models and styles	371
269		

Editor Note

Experts in SC27/WG3 agree with the editors that, since this document needs to reflect the evaluation requirements arising from the various parts of ISO/IEC 15408 the CD for which have only just been completed, it is inevitable that the 18045 draft will lag behind the 15408 parts and that some editing will be needed when the other parts are complete.

The aim expressed at WG3 meetings is to have the whole set of documents clearly and comfortably support the co-existence of the different ways of using the criteria for evaluations. In particular the document set should support without conflict, contradiction, or interference, ways of providing assurance that accommodate both detailed specification with transparent, conformance checking (generally the iTTC/cPP route), and also the investigative, judgement-based examination. Evaluations generally combine both approaches to different extents, and different balances are currently preferred by different groups of users and schemes.

This document is intended to meet that aim.

Notes for CD1

A new element ACE_CCO.1.6C has been added in this version of 18045 (in order to more clearly specify the requirement for its related work units – previously these were attached to ACE_CCO.1.3C but the connection was not clear or convincing). This new element therefore needs to be added to 15408-3.

Optional SFRs have been removed from 18045 (but will be replaced if discussion on other parts necessitates that step)

Note

ISO/IEC 15408-3 CD1 needs to reflect the updated ASE_REQ.1.9C

The security requirements rationale shall demonstrate that the SFRs (in conjunction with the security objectives for the environment) counter all threats for the TOE.

Edits have introduced a new ACE_CCO.1.6C and then renumbered ACE_CCO.1-3a and ACE_CCO.1-3b as ACE_CCO.1-6 and ACE_CCO.1-7 as its work units. This means that the old ACE_CCO.1-6 is now renumbered as ACE_CCO.1-8.

This requires a future update to part 3 to introduce ACE_CCO.1.6C.

APE_CCL.1.13C addresses only the identification of allowed PP-modules whereas the related Work Unit APE_CCL.1-17 covers more, i.e. subject PP's conformance statement / aspect 'allowed with' other base-PPs. ~~~~This seems to be a mismatch, i.e. in APE_CCL.1.13C the goal and content of Work Unit APE_CCL.1-17 is not covered. – expert text awaited

Clarification to what a PP may claim conformance. Corresponding update of ISO/IEC 15408-1, ISO/IEC 15408-3 and / or ISO/IEC 18045. – deferred to incorporate updates

Check as proposed the new subchapters for AVA_VAN.5, ADV_SPM.1, ADV_TDS.5, ADV_IMP.2, ADV_INT.3, ATE_COV.3 and ATE_FUN.2 for consistency to ISO/IEC 15408-1, ISO/IEC 15408-3 and ISO/IEC 18045 (for the latter one check against the other already existing subchapters in AVA, ADV and ATE). Corresponding update of the subchapters where necessary. – expert check awaited (from authors of AIS 34 in particular)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

ISO/IEC 18045 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 27, IT Security techniques. The identical text of ISO/IEC 15408 is published by the Common Criteria Project Sponsoring Organisations as Common Criteria for Information Technology Security Evaluation. The common XML source for both publications can be found at <http://www.commoncriteriaportal.org/cc/>

This **fourth** edition cancels and replaces the **third** edition (ISO/IEC 18045:-2009), which has been technically revised.

The main changes compared to the previous edition are as follows:

—

347 **Introduction**

348 The target audience for this International Standard is primarily evaluators applying ISO/IEC 15408 and certifiers
349 confirming evaluator actions; evaluation sponsors, developers, PP/ST authors and other parties interested in IT
350 security may be a secondary audience.

351 This International Standard recognises that not all questions concerning IT security evaluation will be answered
352 herein and that further interpretations will be needed. Individual schemes will determine how to handle such
353 interpretations, although these may be subject to mutual recognition agreements. A list of methodology-related
354 activities that may be handled by individual schemes can be found in Annex A.

IT security techniques — Evaluation criteria for IT security Information technology — Security techniques — Methodology for IT security evaluation

1 Scope

This International Standard is a companion document to the “Evaluation criteria for IT security”, ISO/IEC 15408. This International Standard defines the minimum actions to be performed by an evaluator in order to conduct an ISO/IEC 15408 evaluation, using the criteria and evaluation evidence defined in ISO/IEC 15408.

This International Standard defines evaluator actions for ISO/IEC 15408 components where there is agreed guidance. Evaluation activities defined in conformance with ISO/IEC 15408-4 may be used in place of work units within this document provided that this is made clear within the evaluation and certification reports.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15408 (all parts), IT security techniques — *Evaluation criteria for IT security*

3 Terms and definitions

For the purposes of this document, the terms, definitions, symbols and abbreviated terms given in ISO/IEC 15408-1 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at <http://www.electropedia.org/>

— ISO Online browsing platform: available at <http://www.iso.org/obp>

Terms and definitions previously located in this clause of 18045 are now found in ISO/IEC 15408

4 Symbols and abbreviated terms

Symbols and abbreviations have been moved to part 1

5 Overview

5.1 Organisation of this International Standard

Clause 6 defines the conventions used in this International Standard.

386 Clause 7 describes general evaluation tasks with no verdicts associated with them as they do not
387 map to ISO/IEC 15408 evaluator action elements.

388 Clause 8 to 10 address the work necessary for reaching an evaluation result on a PP.

389 Clauses 10 to 0 define the evaluation activities, organised by Assurance Classes.

390 Annex A covers the basic evaluation techniques used to provide technical evidence of evaluation
391 results.

392 Annex B provides an explanation of the Vulnerability Analysis criteria and examples of their
393 application

394 **6 Document Conventions**

395 **6.1 Terminology**

396 Unlike ISO/IEC 15408, where each element maintains the last digit of its identifying symbol for all
397 components within the family, this International Standard may introduce new work units when an
398 ISO/IEC 15408 evaluator action element changes from sub-activity to sub-activity; as a result, the
399 last digit of the work unit's identifying symbol may change although the work unit remains
400 unchanged.

401 Any methodology-specific evaluation work required that is not derived directly from ISO/IEC
402 15408 requirements is termed *task* or *sub-task*.

403 **6.2 Verb usage**

404 All work unit and sub-task verbs are preceded by the auxiliary verb *shall* and by presenting both
405 the verb and the *shall* in ***bold italic*** type face. The auxiliary verb *shall* is used only when the
406 provided text is mandatory and therefore only within the work units and sub-tasks. The work units
407 and sub-tasks contain mandatory activities that the evaluator must perform in order to assign
408 verdicts.

409 Guidance text accompanying work units and sub-tasks gives further explanation on how to apply
410 ISO/IEC 15408 words in an evaluation. The verb usage is in accordance with ISO definitions for
411 these verbs. The auxiliary verb *should* is used when the described method is strongly preferred. All
412 other auxiliary verbs, including *may*, are used where the described method(s) is allowed but is
413 neither recommended nor strongly preferred; it is merely explanation.

414 The verbs *check*, *examine*, *report* and *record* are used with a precise meaning within this part of this
415 International Standard and the Clause 3 should be referenced for their definitions.

416 **6.3 General evaluation guidance**

417 Material that has applicability to more than one sub-activity is collected in one place. Guidance
418 whose applicability is widespread (across activities and EALs) has been collected into Annex A.
419 Guidance that pertains to multiple sub-activities within a single activity has been provided in the
420 introduction to that activity. If guidance pertains to only a single sub-activity, it is presented within
421 that sub-activity.

422 **6.4 Relationship between ISO/IEC 15408 and ISO/IEC 18045 structures**

423 There are direct relationships between ISO/IEC 15408 structure (i.e. class, family, component and
424 element) and the structure of this International Standard. Figure 1 illustrates the correspondence
425 between ISO/IEC 15408 constructs of class, family and evaluator action elements and evaluation
426 methodology activities, sub-activities and actions. However, several evaluation methodology work

units may result from the requirements noted in ISO/IEC 15408 developer action and content and presentation elements. Evaluation activities defined in conformance with part 4 of ISO/IEC 15408 may be used in place of work units within this document provided that this is made clear within the evaluation and certification reports

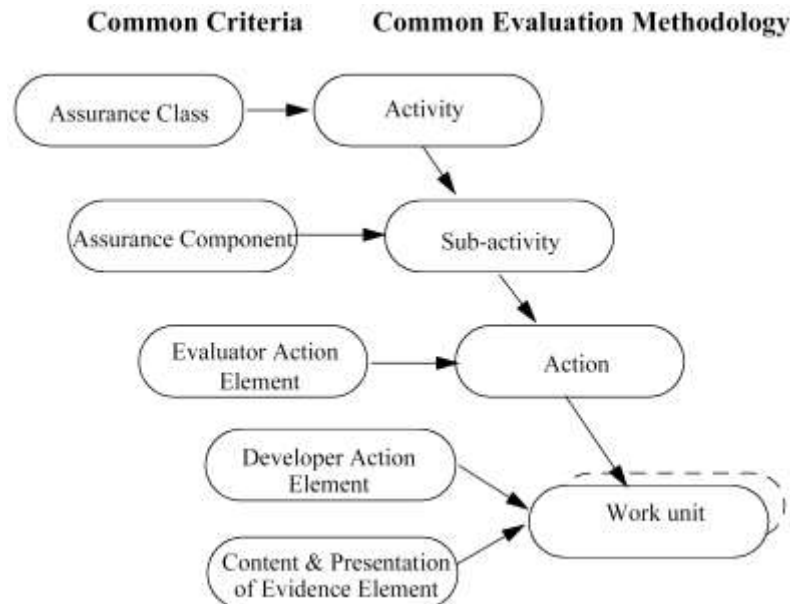


Figure 1 — Mapping of ISO/IEC 15408 and ISO/IEC 18045 structures

7 Evaluation process and related tasks

7.1 Introduction

This clause provides an overview of the evaluation process and defines the tasks an evaluator is intended to perform when conducting an evaluation.

Each evaluation, whether of a PP or TOE (including ST), follows the same process, and has four evaluator tasks in common: the input task, the output task, the evaluation sub-activities, and the demonstration of the technical competence to the evaluation authority task.

The input task and the output tasks, which are related to management of evaluation evidence and to report generation, are entirely described in this clause. Each task has associated sub-tasks that apply to, and are normative for all ISO/IEC 15408 evaluations (evaluation of a PP or a TOE).

The evaluation sub-activities are only introduced in this clause, and fully described in the following clauses.

In contrast to the evaluation sub-activities, input and output tasks have no verdicts associated with them as they do not map to ISO/IEC 15408 evaluator action elements; they are performed in order to ensure conformance with the universal principles and to comply with this International Standard.

The demonstration of the technical competence to the evaluation authority task may be fulfilled by the evaluation authority analysis of the output tasks results, or may include the demonstration by the evaluators of their understanding of the inputs for the evaluation sub-activities. This task has no associated evaluator verdict, but has an evaluator authority verdict. The detailed criteria to pass this task are left to the discretion of the evaluation authority, as noted in Annex A.5.

7.2 Evaluation process overview

7.2.1 Objectives

This subclause presents the general model of the methodology and identifies:

- a) roles and responsibilities of the parties involved in the evaluation process;
- b) the general evaluation model.

7.2.2 Responsibilities of the roles

The general model defines the following roles: sponsor, developer, evaluator and evaluation authority.

The sponsor is responsible for requesting and supporting an evaluation. This means that the sponsor establishes the different agreements for the evaluation (e.g. commissioning the evaluation). Moreover, the sponsor is responsible for ensuring that the evaluator is provided with the evaluation evidence.

The developer produces the TOE and is responsible for providing the evidence required for the evaluation (e.g. training, design information), on behalf of the sponsor.

The evaluator performs the evaluation tasks required in the context of an evaluation: the evaluator receives the evaluation evidence from the developer on behalf of the sponsor or directly from the sponsor, performs the evaluation sub-activities and provides the results of the evaluation assessment to the evaluation authority.

The evaluation authority establishes and maintains the scheme, monitors the evaluation conducted by the evaluator, and issues certification/validation reports as well as certificates based on the evaluation results provided by the evaluator.

7.2.3 Relationship of roles

To prevent undue influence from improperly affecting an evaluation, some separation of roles is required. This implies that the roles described above are fulfilled by different entities, except that the roles of developer and sponsor may be satisfied by a single entity.

Moreover, some evaluations (e.g. EAL1 evaluation) may not require the developer to be involved in the project. In this case, it is the sponsor who provides the TOE to the evaluator and who generates the evaluation evidence.

7.2.4 General evaluation model

The evaluation process consists of the evaluator performing the evaluation input task, the evaluation output task and the evaluation sub-activities. Figure 2 provides an overview of the relationship between these tasks and sub-activities.

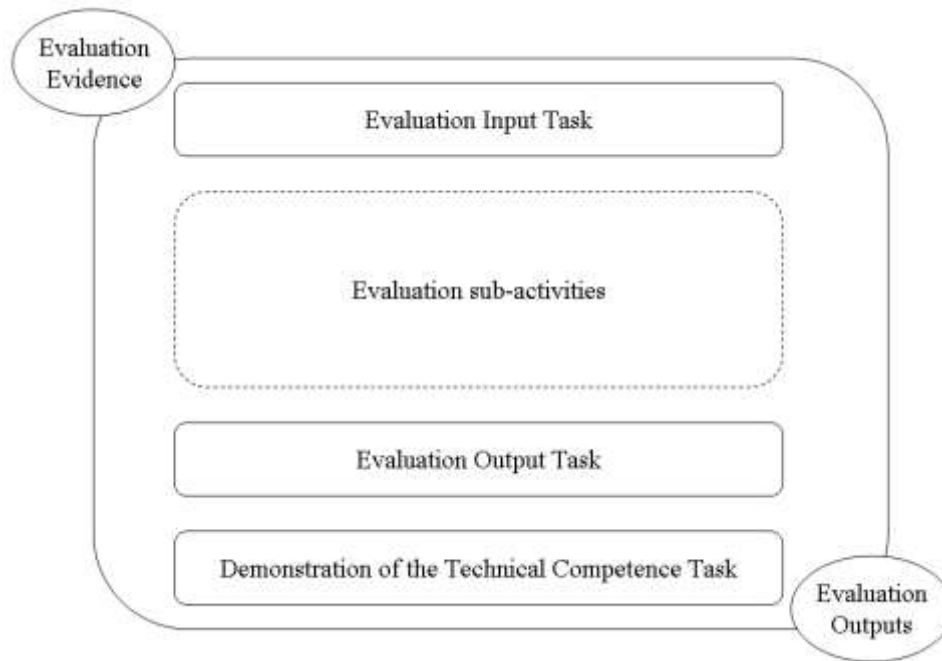


Figure 2 — Generic evaluation model

The evaluation process may be preceded by a preparation phase where initial contact is made between the sponsor and the evaluator. The work that is performed and the involvement of the different roles during this phase may vary. It is typically during this step that the evaluator performs a feasibility analysis to assess the likelihood of a successful evaluation.

7.2.5 Evaluator verdicts

The evaluator assigns verdicts to the requirements of ISO/IEC 15408 and not to those of this International Standard. The most granular ISO/IEC 15408 structure to which a verdict is assigned is the evaluator action element (explicit or implied). A verdict is assigned to an applicable ISO/IEC 15408 evaluator action element as a result of performing the corresponding evaluation methodology action and its constituent work units. Finally, an evaluation result is assigned, as described in ISO/IEC 15408-1, Clause 9, **Evaluation results**.

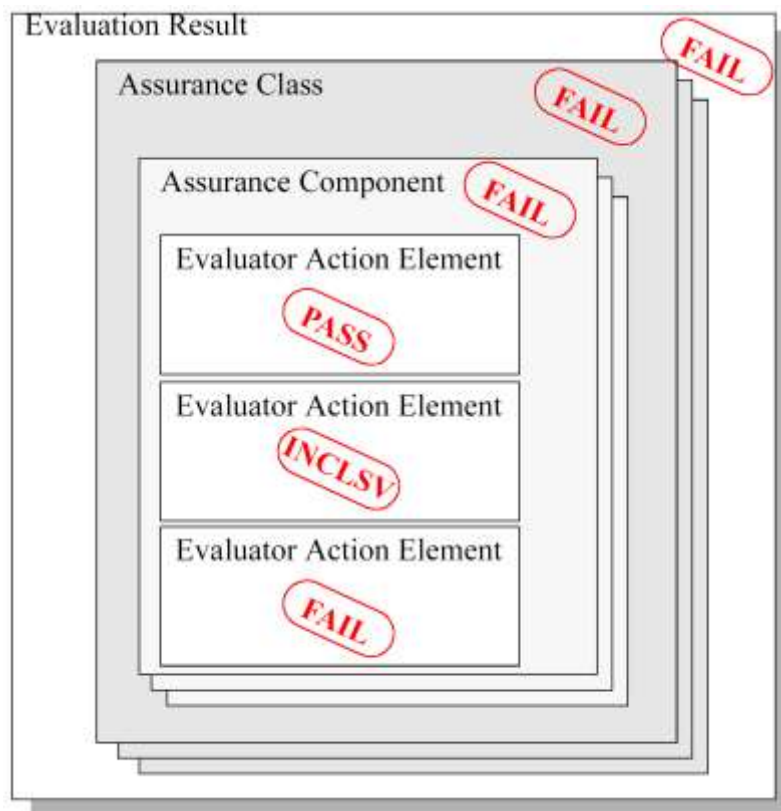


Figure 3 — Example of the verdict assignment rule

This International Standard recognises three mutually exclusive verdict states:

- a) Conditions for a *pass* verdict are defined as an evaluator completion of ISO/IEC 15408 evaluator action element and determination that the requirements for the PP, ST or TOE under evaluation are met. The conditions for passing the element are defined as:
 - 1) the constituent work units of the related evaluation methodology action, and;
 - 2) all evaluation evidence required for performing these work units is coherent, that is it can be fully and completely understood by the evaluator, and
 - 3) all evaluation evidence required for performing these work units does not have any obvious internal inconsistencies or inconsistencies with other evaluation evidence. Note that obvious means here that the evaluator discovers this inconsistency while performing the work units: the evaluator should not undertake a full consistency analysis across the entire evaluation evidence every time a work unit is performed.
- b) Conditions for a *fail* verdict are defined as an evaluator completion of ISO/IEC 15408 evaluator action element and determination that the requirements for the PP, ST, or TOE under evaluation are not met, or that the evidence is incoherent, or an obvious inconsistency in the evaluation evidence has been found;
- c) All verdicts are initially *inconclusive* and remain so until either a *pass* or *fail* verdict is assigned.

The overall verdict is *pass* if and only if all the constituent verdicts are also *pass*. In the example illustrated in Figure 3, if the verdict for one evaluator action element is *fail* then the verdicts for the corresponding assurance component, assurance class, and overall verdict are also *fail*.

522 7.3 Evaluation input task

523 7.3.1 Objectives

524 The objective of this task is to ensure that the evaluator has available the correct version of the
525 evaluation evidence necessary for the evaluation and that it is adequately protected. Otherwise, the
526 technical accuracy of the evaluation cannot be assured, nor can it be assured that the evaluation is
527 being conducted in a way to provide repeatable and reproducible results.

528 7.3.2 Application notes

529 The responsibility to provide all the required evaluation evidence lies with the sponsor. However,
530 most of the evaluation evidence is likely to be produced and supplied by the developer, on behalf of
531 the sponsor.

532 **7.3.2.1 Since the assurance requirements apply to the entire TOE, all evaluation evidence pertaining to all**
533 **parts of the TOE is to be made available to the evaluator. The scope and required content of such**
534 **evaluation evidence is independent of the level of control that the developer has over each of the**
535 **parts of the TOE. For example, if design is required, then the Objectives**

536 **7.3.2.1** The objectives of this sub-activity are to determine whether the formal security policy model of
537 the TSF clearly and consistently describes the rules and characteristics of the security policies
538 and whether this description corresponds with the description of security functions in the
539 functional specification.

540 7.3.2.1 Input

541 7.3.2.1 The evaluation evidence for this sub-activity is:

542 7.3.2.1 the ST;

543 7.3.2.1 the functional specification;

544 7.3.2.1 formal security policy model (ADV_SPM.1.1D);

545 7.3.2.1 formal proof of correspondence between the model and any formal functional specification
546 (ADV_SPM.1.3D);

547 7.3.2.1 demonstration of correspondence between the model and the functional specification
548 (ADV_SPM.1.4D).

549 7.3.2.1 Application notes

550 7.3.2.1 This activity applies to cases where the developer has provided a formal security policy model of
551 the TOE.

552 7.3.2.1 A formal TOE security policy model is a representation of the rules (synonymously termed
553 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
554 Their formal counterparts are called security properties and security features, respectively. The
555 representation includes but is not limited to algebraic specifications, finite state machines and
556 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
557 model is accompanied by an informal interpretation explaining how the rules and characteristics
558 are mapped to the respective properties and features.

559 7.3.2.1 The creation of a formal security policy model helps to identify and eliminate ambiguous,
560 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
561 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
562 of how well the developer has understood the security functionality being implemented and

563 whether there are inconsistencies between the security requirements and the TOE design. The
564 confidence in the model is accompanied by a proof that it contains no inconsistencies.

565 **7.3.2.1** A formal security model is a precise formal presentation of the important aspects of
566 security and their relationship to the behaviour of the TOE; it identifies the set of rules
567 (principles) that defines the TOE security policy and the set of practises (characteristics) that
568 regulates how the TSF manages, protects, and otherwise controls the system resources. The
569 model includes the set of restrictions and properties that specify how information and computing
570 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
571 engineering arguments showing that these restrictions and properties play a key role in the
572 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
573 as well as ancillary text to explain the model and to provide it with context. The security
574 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
575 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

576 **7.3.2.1** The Security Policy Model of the TOE is informally abstracted from its realisation by
577 considering the proposed security requirements of the ST. The informal abstraction is taken to be
578 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
579 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
580 are always prone to fallacies; especially if relationships among subjects, objects and operations
581 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
582 characteristics of the security policy model are mapped to respective properties and features
583 within some formal system, whose rigour and strength can afterwards be used to obtain the
584 security properties by means of theorems and formal proof.

585 **7.3.2.1** While the term "formal security policy model" is used in academic circles, the CC's
586 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
587 claimed. Therefore, the formal security policy model is merely a formal representation of the set
588 of SFRs being claimed.

589 **7.3.2.1** The term security policy has traditionally been associated with only access control
590 policies, whether label-based (mandatory access control) or user-based (discretionary access
591 control). However, a security policy is not limited to access control; there are also audit policies,
592 identification policies, authentication policies, encryption policies, management policies, and any
593 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
594 contains an assignment for identifying these policies that are formally modelled.

595 **7.3.2.1** It is recognized that not all policies can be formally modelled for all TOEs. This is
596 because either a given policy can not be formally modelled in the otherwise well suited
597 framework, or because the nature of the TOE renders impossible the modelling of policies that
598 would otherwise be possible to model.

599 **7.3.2.1 Action ADV_SPM.1.1E**

600 **7.3.2.1 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
601 *text as required, and identify the security policies of the TSF that are modelled.*

602 **7.3.2.1 Work unit ADV_SPM.1-1**

603 **7.3.2.1** The evaluator ***shall examine the TOE security policy model to determine that it is***
604 **written in a formal style.**

605 **7.3.2.1** The evaluator identifies the formal framework upon which the TOE security policy
606 model is based and ensures that it is founded on well established mathematical concepts. **They**
607 **also identify the security properties and features addressed in the application notes and ensure**
608 **the formalization of at least one security policy.**

609 **7.3.2.1** For guidance on formal methods refer to ISO/IEC 15408-3

610 **7.3.2.1 Work unit ADV_SPM.1-2**

611 **7.3.2.1** The evaluator *shall examine the TOE security policy model to determine that it*
 612 contains all necessary informal explanatory text.

613 **7.3.2.1** Supporting narrative descriptions are necessary for all parts of the model (for example,
 614 to make clear the meaning of any formal notation and how they are used) including the security
 615 properties and features.

616 **7.3.2.1 Work unit ADV_SPM.1-3**

617 **7.3.2.1** The evaluator *shall examine the TOE security policy model to determine that all*
 618 security policies of the TSF are identified that are modelled.

619 **7.3.2.1** The evaluator determines whether the SPM identifies the security policies for which a
 620 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 621 of the modelled policies.

622 **7.3.2.1** The evaluator determines whether the list of security policies identified by the SPM is
 623 consistent with the assignment of ADV_SPM.1.1D in the ST.

624 **7.3.2.1** The evaluator determines whether for each security policy identified by the SPM a
 625 model is in fact provided.

626 **7.3.2.1 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 627 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 628 *not secure.*

629 **7.3.2.1 Work unit ADV_SPM.1-4**

630 **7.3.2.1** The evaluator *shall examine the principles and characteristics of the security policies*
 631 to determine that the modelled security behaviour of the TOE is clearly articulated.

632 **7.3.2.1** The security policies are expressed in terms of security principles (rules) which are
 633 modelled by security properties and define the secure state of the TOE. For example, a model
 634 based on state transitions could describe the security policies in terms of principles of its states,
 635 identify its initial state, and define what it means to be a secure state.

636 **7.3.2.1** The evaluator determines that the security policies are reflected within their formal
 637 counterparts of the TSP model.

638 **7.3.2.1** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 639 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 640 resources including attributes and conditions of the TOE) which are modelled by security
 641 features. For example, a model based on state transitions could describe the characteristics as
 642 possible actions in each secure state in a level of detail sufficient to decide into which state the
 643 TOE will be transformed by that action.

644 **7.3.2.1** Together the security principles and characteristics describe the entire security posture
 645 of the TOE.

646 **7.3.2.1** In the context of a formal TOE security policy model the security behaviour is
 647 considered to be clearly articulated only if an adequate mapping from principles and
 648 characteristics to their respective formal counterparts properties and features has been given.
 649 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 650 detailed enough to allow for correct identification of all security objectives and the relation to the
 651 security environment.

- 652 **7.3.2.1** The above condition for clear articulation is necessary but not sufficient. An informal
 653 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 654 must be provided in order to make clear their intended meaning.
- 655 **7.3.2.1 Work unit ADV_SPM.1-5**
- 656 **7.3.2.1** The evaluator *shall examine the TOE security policy model rationale to determine that*
 657 *it formally proves that the security features enforce the security properties.*
- 658 **7.3.2.1** To determine the enforcement, the evaluator considers the security properties and the
 659 security features and verifies that the arguments used in the proof are valid. The proof of
 660 correspondence between the security properties and the security features shall be formal.
- 661 **7.3.2.1** The validity of the security properties shall mean that the TOE is in a secure state. By
 662 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 663 state.
- 664 **7.3.2.1 Work unit ADV_SPM.1-6**
- 665 **7.3.2.1** The evaluator *shall examine the TOE security policy model rationale to determine that*
 666 *it proves the internal consistency of the TOE security policy model.*
- 667 **7.3.2.1** The proof shall show the absence of contradictions within the TOE security policy
 668 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 669 used in the proof are valid.
- 670 **7.3.2.1** Since the TOE security policy model is formal, the proof of its internal consistency shall
 671 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 672 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 673 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 674 security policy model that prove the internal consistency by means of a combination with generic
 675 arguments of the formal framework.
- 676 **7.3.2.1 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 677 *specification shall be at the correct level of formality.*
- 678 **7.3.2.1 Work unit ADV_SPM.1-7**
- 679 **7.3.2.1** The evaluator *shall examine the correspondence between the model and the functional*
 680 *specification to determine that a semiformal demonstration of correspondence between the*
 681 *model and any semiformal functional specification is provided.*
- 682 **7.3.2.1** This work unit is only applicable to a semiformal presentation of the functional
 683 specification, which is required by ADV_FSP.5.2C.
- 684 **7.3.2.1** A semiformal correspondence is one that results from a structured approach with a
 685 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 686 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 687 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 688 **7.3.2.1** For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
 689 methods’.
- 690 **7.3.2.1 Work unit ADV_SPM.1-8**
- 691 **7.3.2.1** The evaluator *shall examine the correspondence between the model and the functional*
 692 *specification to determine that a formal proof of correspondence between the model and any*
 693 *formal functional specification is provided.*

- 694 **7.3.2.1** This work unit is only applicable to a formal presentation of the functional specification,
695 which is required by ADV_FSP.6.2D.
- 696 **7.3.2.1** There should be a formal proof of correspondence between the model and any formal
697 functional specification.
- 698 **7.3.2.1** The formal proof of correspondence removes all subjective interpretations of its terms
699 by enlisting well-established mathematical concepts to define the syntax and semantics of the
700 formal notation and uses rules that support logical reasoning. The security features within the
701 TOE (which are identified in the formal TSP model) are expressed in a formal specification
702 language and shown to be satisfied by the formal specification.
- 703 **7.3.2.1** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 704 **7.3.2.1 ADV_SPM.1.4C** *The correspondence shall show that the functional*
705 *specification is consistent and complete with respect to the model.*
- 706 **7.3.2.1 Work unit ADV_SPM.1-9**
- 707 **7.3.2.1** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
708 **TSF interfaces (as articulated in the functional specification) is complete with respect to the**
709 **behaviour modelled by the security features.**
- 710 **7.3.2.1** The term “correspondence” here means both the formal proof of correspondence
711 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
712 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 713 **7.3.2.1** In determining completeness of the correspondence, the evaluator considers the
714 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
715 features of the TSP model. The demonstration should show that all characteristics belonging to
716 policies that are required to be modelled have an associated feature description in the TOE
717 security policy model, and that each feature of the TSP model does occur in the mapping.
- 718 **7.3.2.1** Abstention from formally modelling TSFI behaviour always calls for justification on the
719 developer’s side (also confer the application notes above).
- 720 **7.3.2.1 Work unit ADV_SPM.1-10**
- 721 **7.3.2.1** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
722 **TSF interfaces (as articulated in the functional specification) is consistent with respect to the**
723 **behaviour modelled by the security features.**
- 724 **7.3.2.1** The term “correspondence” here means both the formal proof of correspondence
725 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
726 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 727 **7.3.2.1** The meaning of consistency reflects the conventional understanding in contrast to the
728 internal consistency concept of work unit ADV_SPM.1-6.
- 729 **7.3.2.1** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
730 security features established in the preceding work unit and verifies that the correspondence
731 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
732 behaviour.
- 733 **7.3.2.1** For example, if TSFI behaviour dealt with access management on the granularity of
734 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
735 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access

736 management for groups of users, then a TSP model describing the security behaviour of the TOE
737 in terms of individual users would also not be consistent.

738 **7.3.2.1** As another example, if remote untrusted users had to pass more stringent
739 authentication procedures than administrators whose only point of access were within a
740 physically-protected area, then this difference in authentication procedures had to be reflected in
741 the security features.

742 **7.3.2.1** TOE design (ADV_TDS) requirements will apply to all subsystems that are part of the
743 TSF. In addition, assurance requirements that call for procedures to be in place (for example, CM
744 capabilities (ALC_CMC) and Delivery (ALC_DEL)) will also apply to the entire TOE (including any
745 part produced by another developer).

746 It is recommended that the evaluator, in conjunction with the sponsor, produce an index to
747 required evaluation evidence. This index may be a set of references to the documentation. This
748 index should contain enough information (e.g. a brief summary of each document, or at least an
749 explicit title, indication of the subclauses of interest) to help the evaluator to find easily the
750 required evidence.

751 It is the information contained in the evaluation evidence that is required, not any particular
752 document structure. Evaluation evidence for a sub-activity may be provided by separate
753 documents, or a single document may satisfy several of the input requirements of a sub-activity.

754 The evaluator requires stable and formally-issued versions of evaluation evidence. However, draft
755 evaluation evidence may be provided during an evaluation, for example, to help an evaluator make
756 an early, informal assessment, but is not used as the basis for verdicts. It may be helpful for the
757 evaluator to see draft versions of particular appropriate evaluation evidence, such as:

758 a) test documentation, to allow the evaluator to make an early assessment of tests and test
759 procedures;

760 b) design documents, to provide the evaluator with background for understanding the TOE
761 design;

762 c) source code or hardware drawings, to allow the evaluator to assess the application of the
763 developer's standards.

764 Draft evaluation evidence is more likely to be encountered where the evaluation of a TOE is
765 performed concurrently with its development. However, it may also be encountered during the
766 evaluation of an already-developed TOE where the developer has had to perform additional work
767 to address a problem identified by the evaluator (e.g. to correct an error in design or
768 implementation) or to provide evaluation evidence of security that is not provided in the existing
769 documentation (e.g. in the case of a TOE not originally developed to meet the requirements of
770 ISO/IEC 15408).

771 **7.3.3 Management of evaluation evidence sub-task**

772 **7.3.3.1 Configuration control**

773 The evaluator **shall perform** configuration control of the evaluation evidence.

774 ISO/IEC 15408 implies that the evaluator is able to identify and locate each item of evaluation
775 evidence after it has been received and is able to determine whether a specific version of a
776 document is in the evaluator's possession.

777 The evaluator **shall protect** the evaluation evidence from alteration or loss while it is in the
778 evaluator's possession.

779 **7.3.3.2 Disposal**

780 Schemes may wish to control the disposal of evaluation evidence at the conclusion of an evaluation.
781 The disposal of the evaluation evidence should be achieved by one or more of:

- 782 a) returning the evaluation evidence;
- 783 b) archiving the evaluation evidence;
- 784 c) destroying the evaluation evidence.

785 **7.3.3.3 Confidentiality**

786 An evaluator may have access to sponsor and developer commercially-sensitive information (e.g.
787 TOE design information, specialist tools), and may have access to nationally-sensitive information
788 during the course of an evaluation. Schemes may wish to impose requirements for the evaluator to
789 maintain the confidentiality of the evaluation evidence. The sponsor and evaluator may mutually
790 agree to additional requirements as long as these are consistent with the scheme.

791 Confidentiality requirements affect many aspects of evaluation work, including the receipt,
792 handling, storage and disposal of evaluation evidence.

793 **7.4 Evaluation sub-activities**

794 The evaluation sub-activities vary depending whether it is a PP or a TOE evaluation. Moreover, in
795 the case of a TOE evaluation, the sub-activities depend upon the selected assurance requirements.

796 **7.5 Evaluation output task**

797 **7.5.1 Objectives**

798 The objective of this subclause is to describe the Observation Report (OR) and the Evaluation
799 Technical Report (ETR). Schemes may require additional evaluator reports such as reports on
800 individual units of work, or may require additional information to be contained in the OR and the
801 ETR. This International Standard does not preclude the addition of information into these reports
802 as this International Standard specifies only the minimum information content.

803 Consistent reporting of evaluation results facilitates the achievement of the universal principle of
804 repeatability and reproducibility of results. The consistency covers the type and the amount of
805 information reported in the ETR and OR. ETR and OR consistency among different evaluations is
806 the responsibility of the evaluation authority.

807 The evaluator performs the two following sub-tasks in order to meet the requirements of this
808 International Standard for the information content of reports:

- 809 a) write OR sub-task (if needed in the context of the evaluation);
- 810 b) write ETR sub-task.

811 **7.5.2 Management of evaluation outputs**

812 The evaluator delivers the ETR to the evaluation authority, as well as any ORs as they become
813 available. Requirements for controls on handling the ETR and ORs are established by the scheme
814 which may include delivery to the sponsor or developer. The ETR and ORs may include sensitive or
815 proprietary information and may need to be sanitised before they are given to the sponsor.

7.5.3 Application notes

In this version of this International Standard, the requirements for the provision of evaluator evidence to support re-evaluation and re-use have not been explicitly stated. Where information for re-evaluation or re-use is required by the sponsor, the scheme under which the evaluation is being performed should be consulted.

7.5.4 Write OR sub-task

ORs provide the evaluator with a mechanism to request a clarification (e.g. from the evaluation authority on the application of a requirement) or to identify a problem with an aspect of the evaluation.

In the case of a fail verdict, the evaluator **shall provide** an OR to reflect the evaluation result. Otherwise, the evaluator may use ORs as one way of expressing clarification needs.

For each OR, the evaluator **shall report** the following:

- a) the identifier of the PP or TOE evaluated;
- b) the evaluation task/sub-activity during which the observation was generated;
- c) the observation;
- d) the assessment of its severity (e.g. implies a fail verdict, holds up progress on the evaluation, requires a resolution prior to evaluation being completed);
- e) the identification of the organisation responsible for resolving the issue;
- f) the recommended timetable for resolution;
- g) the assessment of the impact on the evaluation of failure to resolve the observation.

The intended audience of an OR and procedures for handling the report depend on the nature of the report's content and on the scheme. Schemes may distinguish different types of ORs or define additional types, with associated differences in required information and distribution (e.g. evaluation ORs to evaluation authorities and sponsors).

7.5.5 Write ETR sub-task

7.5.5.1 Objectives

The evaluator **shall provide** an ETR to present technical justification of the verdicts.

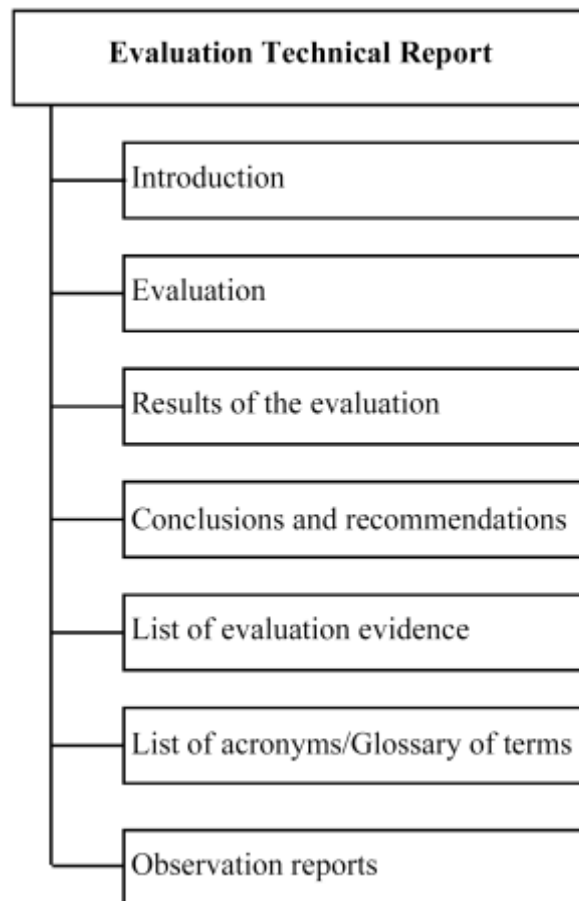
This International Standard defines the ETR's minimum content requirement; however, schemes may specify additional content and specific presentational and structural requirements. For instance, schemes may require that certain introductory material (e.g. disclaimers and copyright Clauses) be reported in the ETR.

The reader of the ETR is assumed to be familiar with general concepts of information security, ISO/IEC 15408, this International Standard, evaluation approaches and IT.

The ETR supports the evaluation authority to confirm that the evaluation was done to the required standard, but it is anticipated that the documented results may not provide all of the necessary information, so additional information specifically requested by the scheme may be necessary. This aspect is outside the scope of this International Standard.

853 7.5.5.2 ETR for a PP Evaluation

854 This Subclause describes the minimum content of the ETR for a PP evaluation. The contents of the
 855 ETR are portrayed in Figure 4; this figure may be used as a guide when constructing the structural
 856 outline of the ETR document.



857

858 **Figure 4 —ETR information content for a PP evaluation**

859 7.5.5.2.1 Introduction

860 The evaluator **shall report** evaluation scheme identifiers.

861 Evaluation scheme identifiers (e.g. logos) are the information required to unambiguously identify
 862 the scheme responsible for the evaluation oversight.

863 The evaluator **shall report** ETR configuration control identifiers.

864 The ETR configuration control identifiers contain information that identifies the ETR (e.g. name,
 865 date and version number).

866 The evaluator **shall report** PP configuration control identifiers.

867 PP configuration control identifiers (e.g. name, date and version number) are required to identify
 868 what is being evaluated in order for the evaluation authority to verify that the verdicts have been
 869 assigned correctly by the evaluator.

870 The evaluator **shall report** the identity of the developer.

871 The identity of the PP developer is required to identify the party responsible for producing the PP.

872 The evaluator **shall report** the identity of the sponsor.

873 The identity of the sponsor is required to identify the party responsible for providing evaluation
874 evidence to the evaluator.

875 The evaluator **shall report** the identity of the evaluator.

876 The identity of the evaluator is required to identify the party performing the evaluation and
877 responsible for the evaluation verdicts.

878 **7.5.5.2.2 Evaluation**

879 The evaluator **shall report** the evaluation methods, techniques, tools and standards used.

880 The evaluator references the evaluation criteria, methodology and interpretations used to evaluate
881 the PP.

882 The evaluator **shall report** any constraints on the evaluation, constraints on the handling of
883 evaluation results and assumptions made during the evaluation that have an impact on the
884 evaluation results.

885 The evaluator may include information in relation to legal or statutory aspects, organisation,
886 confidentiality, etc.

887 **7.5.5.2.3 Results of the evaluation**

888 The evaluator **shall report** a verdict and a supporting rationale for each assurance component that
889 constitutes an APE activity, as a result of performing the corresponding evaluation methodology
890 action and its constituent work units.

891 The rationale justifies the verdict using ISO/IEC 15408, this International Standard, any
892 interpretations and the evaluation evidence examined and shows how the evaluation evidence
893 does or does not meet each aspect of the criteria. It contains a description of the work performed,
894 the method used, and any derivation of results. The rationale may provide detail to the level of an
895 evaluation methodology work unit.

896 **7.5.5.2.4 Conclusions and recommendations**

897 The evaluator **shall report** the conclusions of the evaluation, in particular the overall verdict as
898 defined in ISO/IEC 15408-1 Clause 12, Evaluation results, and determined by application of the
899 verdict assignment described in 7.2.5.

900 The evaluator provides recommendations that may be useful for the evaluation authority. These
901 recommendations may include shortcomings of the PP discovered during the evaluation or
902 mention of features which are particularly useful.

903 **7.5.5.2.5 List of evaluation evidence**

904 The evaluator **shall report** for each item of evaluation evidence the following information:

905 — the issuing body (e.g. the developer, the sponsor);

906 — the title;

907 — the unique reference (e.g. issue date and version number).

908 **7.5.5.2.6 List of acronyms/Glossary of terms**

909 The evaluator ***shall report*** any acronyms or abbreviations used in the ETR.

910 Glossary definitions already defined by ISO/IEC 15408 or by this International Standard need not
911 be repeated in the ETR.

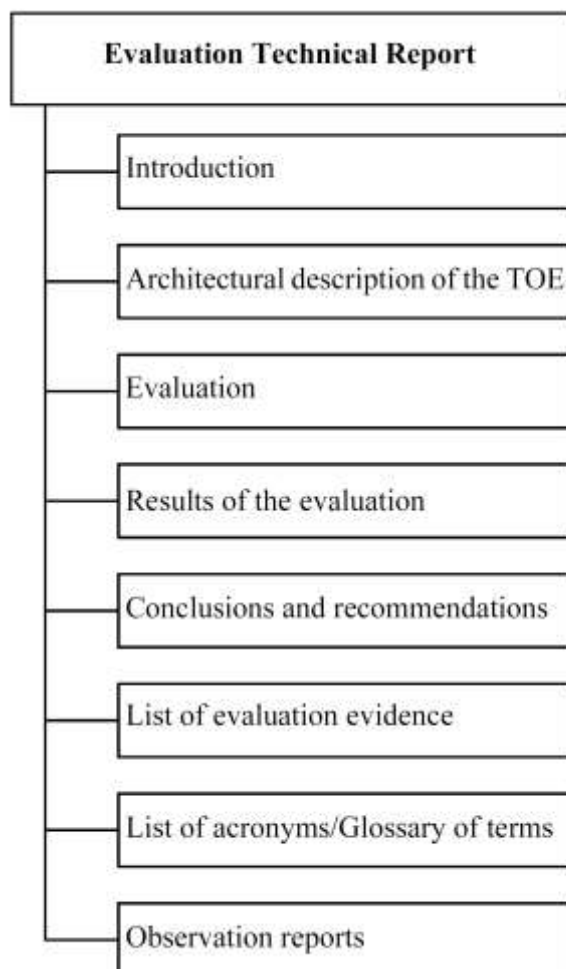
912 **7.5.5.2.7 Observation reports**

913 The evaluator ***shall report*** a complete list that uniquely identifies the ORs raised during the
914 evaluation and their status.

915 For each OR, the list should contain its identifier as well as its title or a brief summary of its content.

916 **7.5.5.3 ETR for a TOE Evaluation**

917 This Subclause describes the minimum content of the ETR for a TOE evaluation. The contents of the
918 ETR are portrayed in Figure 5; this figure may be used as a guide when constructing the structural
919 outline of the ETR document.



920

921 **Figure 5 — ETR information content for a TOE evaluation**

922 **7.5.5.3.1 Introduction**

923 The evaluator ***shall report*** evaluation scheme identifiers.

- 924 Evaluation scheme identifiers (e.g. logos) are the information required to unambiguously identify
925 the scheme responsible for the evaluation oversight.
- 926 The evaluator **shall report** ETR configuration control identifiers.
- 927 The ETR configuration control identifiers contain information that identifies the ETR (e.g. name,
928 date and version number).
- 929 The evaluator **shall report** ST and TOE configuration control identifiers.
- 930 ST and TOE configuration control identifiers identify what is being evaluated in order for the
931 evaluation authority to verify that the verdicts have been assigned correctly by the evaluator.
- 932 If the ST claims that the TOE conforms to the requirements of one or more PPs, the ETR shall
933 report the reference of the corresponding PPs.
- 934 The PPs reference contains information that uniquely identifies the PPs (e.g. title, date, and version
935 number).
- 936 The evaluator **shall report** the identity of the developer.
- 937 The identity of the TOE developer is required to identify the party responsible for producing the
938 TOE.
- 939 The evaluator **shall report** the identity of the sponsor.
- 940 The identity of the sponsor is required to identify the party responsible for providing evaluation
941 evidence to the evaluator.
- 942 The evaluator **shall report** the identity of the evaluator.
- 943 The identity of the evaluator is required to identify the party performing the evaluation and
944 responsible for the evaluation verdicts.
- 945 **7.5.5.3.2 Architectural description of the TOE**
- 946 **7.5.5.4 The evaluator shall report a high level description of the TOE and its major components based on**
947 **the evaluation evidence described in ISO/IEC 15408 assurance family entitled Objectives**
- 948 **7.5.5.4** The objectives of this sub-activity are to determine whether the formal security policy model of
949 the TSF clearly and consistently describes the rules and characteristics of the security policies
950 and whether this description corresponds with the description of security functions in the
951 functional specification.
- 952 **7.5.5.4 Input**
- 953 **7.5.5.4** The evaluation evidence for this sub-activity is:
- 954 **7.5.5.4** the ST;
- 955 **7.5.5.4** the functional specification;
- 956 **7.5.5.4** formal security policy model (ADV_SPM.1.1D);
- 957 **7.5.5.4** formal proof of correspondence between the model and any formal functional specification
958 (ADV_SPM.1.3D);

959 **7.5.5.4** demonstration of correspondence between the model and the functional specification
 960 (ADV_SPM.1.4D).

961 **7.5.5.4 Application notes**

962 **7.5.5.4** This activity applies to cases where the developer has provided a formal security policy
 963 model of the TOE.

964 **7.5.5.4** A formal TOE security policy model is a representation of the rules (synonymously
 965 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
 966 terms. Their formal counterparts are called security properties and security features,
 967 respectively. The representation includes but is not limited to algebraic specifications, finite state
 968 machines and logic formalisms strong enough to formally infer the properties from the features.
 969 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
 970 characteristics are mapped to the respective properties and features.

971 **7.5.5.4** The creation of a formal security policy model helps to identify and eliminate
 972 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
 973 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
 974 judgement of how well the developer has understood the security functionality being
 975 implemented and whether there are inconsistencies between the security requirements and the
 976 TOE design. The confidence in the model is accompanied by a proof that it contains no
 977 inconsistencies.

978 **7.5.5.4** A formal security model is a precise formal presentation of the important aspects of
 979 security and their relationship to the behaviour of the TOE; it identifies the set of rules
 980 (principles) that defines the TOE security policy and the set of practises (characteristics) that
 981 regulates how the TSF manages, protects, and otherwise controls the system resources. The
 982 model includes the set of restrictions and properties that specify how information and computing
 983 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
 984 engineering arguments showing that these restrictions and properties play a key role in the
 985 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
 986 as well as ancillary text to explain the model and to provide it with context. The security
 987 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
 988 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

989 **7.5.5.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
 990 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 991 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 992 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 993 are always prone to fallacies; especially if relationships among subjects, objects and operations
 994 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 995 characteristics of the security policy model are mapped to respective properties and features
 996 within some formal system, whose rigour and strength can afterwards be used to obtain the
 997 security properties by means of theorems and formal proof.

998 **7.5.5.4** While the term “formal security policy model” is used in academic circles, the CC's
 999 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
 1000 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 1001 of SFRs being claimed.

1002 **7.5.5.4** The term security policy has traditionally been associated with only access control
 1003 policies, whether label-based (mandatory access control) or user-based (discretionary access
 1004 control). However, a security policy is not limited to access control; there are also audit policies,
 1005 identification policies, authentication policies, encryption policies, management policies, and any
 1006 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 1007 contains an assignment for identifying these policies that are formally modelled.

- 1008 **7.5.5.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
 1009 because either a given policy can not be formally modelled in the otherwise well suited
 1010 framework, or because the nature of the TOE renders impossible the modelling of policies that
 1011 would otherwise be possible to model.
- 1012 **7.5.5.4 Action ADV_SPM.1.1E**
- 1013 **7.5.5.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 1014 *text as required, and identify the security policies of the TSF that are modelled.*
- 1015 **7.5.5.4 Work unit ADV_SPM.1-1**
- 1016 **7.5.5.4** The evaluator ***shall examine the TOE security policy model to determine that it is***
 1017 ***written in a formal style.***
- 1018 **7.5.5.4** The evaluator identifies the formal framework upon which the TOE security policy
 1019 model is based and ensures that it is founded on well established mathematical concepts. **They**
 1020 **also identify the security properties and features addressed in the application notes and ensure**
 1021 **the formalization of at least one security policy.**
- 1022 **7.5.5.4** For guidance on formal methods refer to ISO/IEC 15408-3
- 1023 **7.5.5.4 Work unit ADV_SPM.1-2**
- 1024 **7.5.5.4** The evaluator ***shall examine the TOE security policy model to determine that it***
 1025 ***contains all necessary informal explanatory text.***
- 1026 **7.5.5.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
 1027 to make clear the meaning of any formal notation and how they are used) including the security
 1028 properties and features.
- 1029 **7.5.5.4 Work unit ADV_SPM.1-3**
- 1030 **7.5.5.4** The evaluator ***shall examine the TOE security policy model to determine that all***
 1031 ***security policies of the TSF are identified that are modelled.***
- 1032 **7.5.5.4** The evaluator determines whether the SPM identifies the security policies for which a
 1033 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 1034 of the modelled policies.
- 1035 **7.5.5.4** The evaluator determines whether the list of security policies identified by the SPM is
 1036 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 1037 **7.5.5.4** The evaluator determines whether for each security policy identified by the SPM a
 1038 model is in fact provided.
- 1039 **7.5.5.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 1040 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 1041 *not secure.*
- 1042 **7.5.5.4 Work unit ADV_SPM.1-4**
- 1043 **7.5.5.4** The evaluator ***shall examine the principles and characteristics of the security policies***
 1044 ***to determine that the modelled security behaviour of the TOE is clearly articulated.***
- 1045 **7.5.5.4** The security policies are expressed in terms of security principles (rules) which are
 1046 modelled by security properties and define the secure state of the TOE. For example, a model

- 1047 based on state transitions could describe the security policies in terms of principles of its states,
1048 identify its initial state, and define what it means to be a secure state.
- 1049 **7.5.5.4** The evaluator determines that the security policies are reflected within their formal
1050 counterparts of the TSP model.
- 1051 **7.5.5.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
1052 portions of TOE security functionality managing, protecting, and otherwise controlling the system
1053 resources including attributes and conditions of the TOE) which are modelled by security
1054 features. For example, a model based on state transitions could describe the characteristics as
1055 possible actions in each secure state in a level of detail sufficient to decide into which state the
1056 TOE will be transformed by that action.
- 1057 **7.5.5.4** Together the security principles and characteristics describe the entire security posture
1058 of the TOE.
- 1059 **7.5.5.4** In the context of a formal TOE security policy model the security behaviour is
1060 considered to be clearly articulated only if an adequate mapping from principles and
1061 characteristics to their respective formal counterparts properties and features has been given.
1062 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
1063 detailed enough to allow for correct identification of all security objectives and the relation to the
1064 security environment.
- 1065 **7.5.5.4** The above condition for clear articulation is necessary but not sufficient. An informal
1066 interpretation of all formal concepts (including attributes, predicates and variables, if available)
1067 must be provided in order to make clear their intended meaning.
- 1068 **7.5.5.4 Work unit ADV_SPM.1-5**
- 1069 **7.5.5.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
1070 *it formally proves that the security features enforce the security properties.*
- 1071 **7.5.5.4** To determine the enforcement, the evaluator considers the security properties and the
1072 security features and verifies that the arguments used in the proof are valid. The proof of
1073 correspondence between the security properties and the security features shall be formal.
- 1074 **7.5.5.4** The validity of the security properties shall mean that the TOE is in a secure state. By
1075 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
1076 state.
- 1077 **7.5.5.4 Work unit ADV_SPM.1-6**
- 1078 **7.5.5.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
1079 *it proves the internal consistency of the TOE security policy model.*
- 1080 **7.5.5.4** The proof shall show the absence of contradictions within the TOE security policy
1081 model. In determining the absence of contradictions, the evaluator verifies that the arguments
1082 used in the proof are valid.
- 1083 **7.5.5.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
1084 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
1085 security policy model usually is not possible due to the fundamental nature of formal frameworks.
1086 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
1087 security policy model that prove the internal consistency by means of a combination with generic
1088 arguments of the formal framework.
- 1089 **7.5.5.4 ADV_SPM.1.3C** ***The correspondence between the model and the functional***
1090 ***specification shall be at the correct level of formality.***

- 1091 **7.5.5.4 Work unit ADV_SPM.1-7**
- 1092 **7.5.5.4** The evaluator ***shall examine the correspondence between the model and the functional***
 1093 specification to determine that a semiformal demonstration of correspondence between the
 1094 model and any semiformal functional specification is provided.
- 1095 **7.5.5.4** This work unit is only applicable to a semiformal presentation of the functional
 1096 specification, which is required by ADV_FSP.5.2C.
- 1097 **7.5.5.4** A semiformal correspondence is one that results from a structured approach with a
 1098 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 1099 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 1100 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 1101 **7.5.5.4** For guidance on semiformal methods refer to Annex 3.1.1 ‘**Semiformal and formal**
 1102 methods’.
- 1103 **7.5.5.4 Work unit ADV_SPM.1-8**
- 1104 **7.5.5.4** The evaluator ***shall examine the correspondence between the model and the functional***
 1105 specification to determine that a formal proof of correspondence between the model and any
 1106 formal functional specification is provided.
- 1107 **7.5.5.4** This work unit is only applicable to a formal presentation of the functional specification,
 1108 which is required by ADV_FSP.6.2D.
- 1109 **7.5.5.4** There should be a formal proof of correspondence between the model and any formal
 1110 functional specification.
- 1111 **7.5.5.4** The formal proof of correspondence removes all subjective interpretations of its terms
 1112 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 1113 formal notation and uses rules that support logical reasoning. The security features within the
 1114 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 1115 language and shown to be satisfied by the formal specification.
- 1116 **7.5.5.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 1117 **7.5.5.4 ADV_SPM.1.4C** ***The correspondence shall show that the functional***
 1118 ***specification is consistent and complete with respect to the model.***
- 1119 **7.5.5.4 Work unit ADV_SPM.1-9**
- 1120 **7.5.5.4** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 1121 TSF interfaces (as articulated in the functional specification) is complete with respect to the
 1122 behaviour modelled by the security features.
- 1123 **7.5.5.4** The term “correspondence” here means both the formal proof of correspondence
 1124 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 1125 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 1126 **7.5.5.4** In determining completeness of the correspondence, the evaluator considers the
 1127 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 1128 features of the TSP model. The demonstration should show that all characteristics belonging to
 1129 policies that are required to be modelled have an associated feature description in the TOE
 1130 security policy model, and that each feature of the TSP model does occur in the mapping.
- 1131 **7.5.5.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
 1132 developer’s side (also confer the application notes above).

1133 **7.5.5.4 Work unit ADV_SPM.1-10**

1134 **7.5.5.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 1135 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 1136 behaviour modelled by the security features.

1137 **7.5.5.4** The term “correspondence” here means both the formal proof of correspondence
 1138 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 1139 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

1140 **7.5.5.4** The meaning of consistency reflects the conventional understanding in contrast to the
 1141 internal consistency concept of work unit ADV_SPM.1-6.

1142 **7.5.5.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 1143 security features established in the preceding work unit and verifies that the correspondence
 1144 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 1145 behaviour.

1146 **7.5.5.4** For example, if TSFI behaviour dealt with access management on the granularity of
 1147 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 1148 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 1149 management for groups of users, then a TSP model describing the security behaviour of the TOE
 1150 in terms of individual users would also not be consistent.

1151 **7.5.5.4** As another example, if remote untrusted users had to pass more stringent
 1152 authentication procedures than administrators whose only point of access were within a
 1153 physically-protected area, then this difference in authentication procedures had to be reflected in
 1154 the security features.

1155 **7.5.5.4** TOE design (ADV_TDS), where applicable.

1156 **7.5.5.8 The intent of this Subclause is to characterise the degree of architectural**
 1157 **separation of the major components. If there is no Objectives**

1158 **7.5.5.8** The objectives of this sub-activity are to determine whether the formal security policy model of
 1159 the TSF clearly and consistently describes the rules and characteristics of the security policies
 1160 and whether this description corresponds with the description of security functions in the
 1161 functional specification.

1162 **7.5.5.8 Input**

1163 **7.5.5.8** The evaluation evidence for this sub-activity is:

1164 **7.5.5.8** the ST;

1165 **7.5.5.8** the functional specification;

1166 **7.5.5.8** formal security policy model (ADV_SPM.1.1D);

1167 **7.5.5.8** formal proof of correspondence between the model and any formal functional specification
 1168 (ADV_SPM.1.3D);

1169 **7.5.5.8** demonstration of correspondence between the model and the functional specification
 1170 (ADV_SPM.1.4D).

1171 **7.5.5.8 Application notes**

1172 **7.5.5.8** This activity applies to cases where the developer has provided a formal security policy
1173 model of the TOE.

1174 **7.5.5.8** A formal TOE security policy model is a representation of the rules (synonymously
1175 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
1176 terms. Their formal counterparts are called security properties and security features,
1177 respectively. The representation includes but is not limited to algebraic specifications, finite state
1178 machines and logic formalisms strong enough to formally infer the properties from the features.
1179 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
1180 characteristics are mapped to the respective properties and features.

1181 **7.5.5.8** The creation of a formal security policy model helps to identify and eliminate
1182 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
1183 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
1184 judgement of how well the developer has understood the security functionality being
1185 implemented and whether there are inconsistencies between the security requirements and the
1186 TOE design. The confidence in the model is accompanied by a proof that it contains no
1187 inconsistencies.

1188 **7.5.5.8** A formal security model is a precise formal presentation of the important aspects of
1189 security and their relationship to the behaviour of the TOE; it identifies the set of rules
1190 (principles) that defines the TOE security policy and the set of practises (characteristics) that
1191 regulates how the TSF manages, protects, and otherwise controls the system resources. The
1192 model includes the set of restrictions and properties that specify how information and computing
1193 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
1194 engineering arguments showing that these restrictions and properties play a key role in the
1195 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
1196 as well as ancillary text to explain the model and to provide it with context. The security
1197 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
1198 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

1199 **7.5.5.8** The Security Policy Model of the TOE is informally abstracted from its realisation by
1200 considering the proposed security requirements of the ST. The informal abstraction is taken to be
1201 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
1202 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
1203 are always prone to fallacies; especially if relationships among subjects, objects and operations
1204 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
1205 characteristics of the security policy model are mapped to respective properties and features
1206 within some formal system, whose rigour and strength can afterwards be used to obtain the
1207 security properties by means of theorems and formal proof.

1208 **7.5.5.8** While the term “formal security policy model” is used in academic circles, the CC's
1209 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
1210 claimed. Therefore, the formal security policy model is merely a formal representation of the set
1211 of SFRs being claimed.

1212 **7.5.5.8** The term security policy has traditionally been associated with only access control
1213 policies, whether label-based (mandatory access control) or user-based (discretionary access
1214 control). However, a security policy is not limited to access control; there are also audit policies,
1215 identification policies, authentication policies, encryption policies, management policies, and any
1216 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
1217 contains an assignment for identifying these policies that are formally modelled.

1218 **7.5.5.8** It is recognized that not all policies can be formally modelled for all TOEs. This is
1219 because either a given policy can not be formally modelled in the otherwise well suited

- 1220 framework, or because the nature of the TOE renders impossible the modelling of policies that
1221 would otherwise be possible to model.
- 1222 **7.5.5.8 Action ADV_SPM.1.1E**
- 1223 **7.5.5.8 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
1224 *text as required, and identify the security policies of the TSF that are modelled.*
- 1225 **7.5.5.8 Work unit ADV_SPM.1-1**
- 1226 **7.5.5.8** The evaluator *shall examine the TOE security policy model to determine that it is*
1227 *written in a formal style.*
- 1228 **7.5.5.8** The evaluator identifies the formal framework upon which the TOE security policy
1229 model is based and ensures that it is founded on well established mathematical concepts. **They**
1230 **also identify the security properties and features addressed in the application notes and ensure**
1231 **the formalization of at least one security policy.**
- 1232 **7.5.5.8** For guidance on formal methods refer to ISO/IEC 15408-3
- 1233 **7.5.5.8 Work unit ADV_SPM.1-2**
- 1234 **7.5.5.8** The evaluator *shall examine the TOE security policy model to determine that it*
1235 *contains all necessary informal explanatory text.*
- 1236 **7.5.5.8** Supporting narrative descriptions are necessary for all parts of the model (for example,
1237 to make clear the meaning of any formal notation and how they are used) including the security
1238 properties and features.
- 1239 **7.5.5.8 Work unit ADV_SPM.1-3**
- 1240 **7.5.5.8** The evaluator *shall examine the TOE security policy model to determine that all*
1241 *security policies of the TSF are identified that are modelled.*
- 1242 **7.5.5.8** The evaluator determines whether the SPM identifies the security policies for which a
1243 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
1244 of the modelled policies.
- 1245 **7.5.5.8** The evaluator determines whether the list of security policies identified by the SPM is
1246 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 1247 **7.5.5.8** The evaluator determines whether for each security policy identified by the SPM a
1248 model is in fact provided.
- 1249 **7.5.5.8 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
1250 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
1251 *not secure.*
- 1252 **7.5.5.8 Work unit ADV_SPM.1-4**
- 1253 **7.5.5.8** The evaluator *shall examine the principles and characteristics of the security policies*
1254 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 1255 **7.5.5.8** The security policies are expressed in terms of security principles (rules) which are
1256 modelled by security properties and define the secure state of the TOE. For example, a model
1257 based on state transitions could describe the security policies in terms of principles of its states,
1258 identify its initial state, and define what it means to be a secure state.

- 1259 **7.5.5.8** The evaluator determines that the security policies are reflected within their formal
1260 counterparts of the TSP model.
- 1261 **7.5.5.8** The TOE security behaviour is expressed in terms of security characteristics (i.e.
1262 portions of TOE security functionality managing, protecting, and otherwise controlling the system
1263 resources including attributes and conditions of the TOE) which are modelled by security
1264 features. For example, a model based on state transitions could describe the characteristics as
1265 possible actions in each secure state in a level of detail sufficient to decide into which state the
1266 TOE will be transformed by that action.
- 1267 **7.5.5.8** Together the security principles and characteristics describe the entire security posture
1268 of the TOE.
- 1269 **7.5.5.8** In the context of a formal TOE security policy model the security behaviour is
1270 considered to be clearly articulated only if an adequate mapping from principles and
1271 characteristics to their respective formal counterparts properties and features has been given.
1272 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
1273 detailed enough to allow for correct identification of all security objectives and the relation to the
1274 security environment.
- 1275 **7.5.5.8** The above condition for clear articulation is necessary but not sufficient. An informal
1276 interpretation of all formal concepts (including attributes, predicates and variables, if available)
1277 must be provided in order to make clear their intended meaning.
- 1278 **7.5.5.8 Work unit ADV_SPM.1-5**
- 1279 **7.5.5.8** The evaluator *shall examine the TOE security policy model rationale to determine that*
1280 *it formally proves that the security features enforce the security properties.*
- 1281 **7.5.5.8** To determine the enforcement, the evaluator considers the security properties and the
1282 security features and verifies that the arguments used in the proof are valid. The proof of
1283 correspondence between the security properties and the security features shall be formal.
- 1284 **7.5.5.8** The validity of the security properties shall mean that the TOE is in a secure state. By
1285 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
1286 state.
- 1287 **7.5.5.8 Work unit ADV_SPM.1-6**
- 1288 **7.5.5.8** The evaluator *shall examine the TOE security policy model rationale to determine that*
1289 *it proves the internal consistency of the TOE security policy model.*
- 1290 **7.5.5.8** The proof shall show the absence of contradictions within the TOE security policy
1291 model. In determining the absence of contradictions, the evaluator verifies that the arguments
1292 used in the proof are valid.
- 1293 **7.5.5.8** Since the TOE security policy model is formal, the proof of its internal consistency shall
1294 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
1295 security policy model usually is not possible due to the fundamental nature of formal frameworks.
1296 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
1297 security policy model that prove the internal consistency by means of a combination with generic
1298 arguments of the formal framework.
- 1299 **7.5.5.8 ADV_SPM.1.3C** *The correspondence between the model and the functional*
1300 *specification shall be at the correct level of formality.*

- 1301 **7.5.5.8 Work unit ADV_SPM.1-7**
- 1302 **7.5.5.8** The evaluator *shall examine the correspondence between the model and the*
 1303 *functional specification to determine that a semiformal demonstration of correspondence*
 1304 *between the model and any semiformal functional specification is provided.*
- 1305 **7.5.5.8** This work unit is only applicable to a semiformal presentation of the functional
 1306 specification, which is required by ADV_FSP.5.2C.
- 1307 **7.5.5.8** A semiformal correspondence is one that results from a structured approach with a
 1308 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 1309 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 1310 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 1311 **7.5.5.8** For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
 1312 methods’.
- 1313 **7.5.5.8 Work unit ADV_SPM.1-8**
- 1314 **7.5.5.8** The evaluator *shall examine the correspondence between the model and the functional*
 1315 *specification to determine that a formal proof of correspondence between the model and any*
 1316 *formal functional specification is provided.*
- 1317 **7.5.5.8** This work unit is only applicable to a formal presentation of the functional specification,
 1318 which is required by ADV_FSP.6.2D.
- 1319 **7.5.5.8** There should be a formal proof of correspondence between the model and any formal
 1320 functional specification.
- 1321 **7.5.5.8** The formal proof of correspondence removes all subjective interpretations of its terms
 1322 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 1323 formal notation and uses rules that support logical reasoning. The security features within the
 1324 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 1325 language and shown to be satisfied by the formal specification.
- 1326 **7.5.5.8** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 1327 **7.5.5.8 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 1328 *specification is consistent and complete with respect to the model.*
- 1329 **7.5.5.8 Work unit ADV_SPM.1-9**
- 1330 **7.5.5.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 1331 *TSF interfaces (as articulated in the functional specification) is complete with respect to the*
 1332 *behaviour modelled by the security features.*
- 1333 **7.5.5.8** The term “correspondence” here means both the formal proof of correspondence
 1334 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 1335 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 1336 **7.5.5.8** In determining completeness of the correspondence, the evaluator considers the
 1337 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 1338 features of the TSP model. The demonstration should show that all characteristics belonging to
 1339 policies that are required to be modelled have an associated feature description in the TOE
 1340 security policy model, and that each feature of the TSP model does occur in the mapping.
- 1341 **7.5.5.8** Abstention from formally modelling TSFI behaviour always calls for justification on the
 1342 developer’s side (also confer the application notes above).

1343 **7.5.5.8 Work unit ADV_SPM.1-10**

1344 **7.5.5.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 1345 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 1346 behaviour modelled by the security features.

1347 **7.5.5.8** The term “correspondence” here means both the formal proof of correspondence
 1348 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 1349 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

1350 **7.5.5.8** The meaning of consistency reflects the conventional understanding in contrast to the
 1351 internal consistency concept of work unit ADV_SPM.1-6.

1352 **7.5.5.8** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 1353 security features established in the preceding work unit and verifies that the correspondence
 1354 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 1355 behaviour.

1356 **7.5.5.8** For example, if TSFI behaviour dealt with access management on the granularity of
 1357 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 1358 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 1359 management for groups of users, then a TSP model describing the security behaviour of the TOE
 1360 in terms of individual users would also not be consistent.

1361 **7.5.5.8** As another example, if remote untrusted users had to pass more stringent
 1362 authentication procedures than administrators whose only point of access were within a
 1363 physically-protected area, then this difference in authentication procedures had to be reflected in
 1364 the security features.

1365 **7.5.5.8** TOE design (ADV_TDS) requirement in the ST, this is not applicable and is considered to
 1366 be satisfied.

1367 **7.5.5.11.11 Evaluation**

1368 The evaluator *shall report* the evaluation methods, techniques, tools and standards used.

1369 The evaluator may reference the evaluation criteria, methodology and interpretations used to
 1370 evaluate the TOE or the devices used to perform the tests.

1371 The evaluator *shall report* any constraints on the evaluation, constraints on the distribution of
 1372 evaluation results and assumptions made during the evaluation that have an impact on the
 1373 evaluation results.

1374 The evaluator may include information in relation to legal or statutory aspects, organisation,
 1375 confidentiality, etc.

1376 **7.5.5.11.12 Results of the evaluation**

1377 For each activity on which the TOE is evaluated, the evaluator *shall report*:

1378 — the title of the activity considered;

1379 — a verdict and a supporting rationale for each assurance component that constitutes this
 1380 activity, as a result of performing the corresponding evaluation methodology action and its
 1381 constituent work units.

1382 The rationale justifies the verdict using ISO/IEC 15408, this International Standard, any
 1383 interpretations and the evaluation evidence examined and shows how the evaluation evidence

1384 does or does not meet each aspect of the criteria. It contains a description of the work performed,
 1385 the method used, and any derivation of results. The rationale may provide detail to the level of an
 1386 evaluation methodology work unit.

1387 The evaluator ***shall report*** all information specifically required by a work unit.

1388 For the AVA and ATE activities, work units that identify information to be reported in the ETR have
 1389 been defined.

1390 **7.5.5.11.13 Conclusions and recommendations**

1391 The evaluator ***shall report*** the conclusions of the evaluation, which will relate to whether the TOE
 1392 has satisfied its associated ST, in particular the overall verdict as defined in ISO/IEC 15408-1
 1393 Clause 9, **Evaluation results**, and determined by application of the verdict assignment described in
 1394 7.2.5.

1395 The evaluator provides recommendations that may be useful for the evaluation authority. These
 1396 recommendations may include shortcomings of the IT product discovered during the evaluation or
 1397 mention of features which are particularly useful.

1398 **7.5.5.11.14 List of evaluation evidence**

1399 The evaluator ***shall report*** for each item of evaluation evidence the following information:

1400 — the issuing body (e.g. the developer, the sponsor);

1401 — the title;

1402 — the unique reference (e.g. issue date and version number).

1403 **7.5.5.11.15 List of acronyms/Glossary of terms**

1404 The evaluator ***shall report*** any acronyms or abbreviations used in the ETR.

1405 Glossary definitions already defined by ISO/IEC 15408 or by this International Standard need not
 1406 be repeated in the ETR.

1407 **7.5.5.11.16 Observation reports**

1408 The evaluator ***shall report*** a complete list that uniquely identifies the ORs raised during the
 1409 evaluation and their status.

1410 For each OR, the list should contain its identifier as well as its title or a brief summary of its content.

1411 **8 Class APE: Protection Profile evaluation**

1412 **8.1 Introduction**

1413 This Clause describes the evaluation of a PP. The requirements and methodology for PP evaluation
 1414 are identical for each PP evaluation, regardless of the EAL (or other set of assurance requirements)
 1415 that is claimed in the PP. The evaluation methodology in this Clause is based on the requirements
 1416 on the PP as specified in ISO/IEC 15408-3 class APE.

1417 This Clause should be used in conjunction with Annexes **A**, **B** and **C**, **Guidance for Operations** in
 1418 ISO/IEC 15408-1, as these Annexes clarify the concepts here and provide many examples.

8.2 Application notes

8.2.1 Re-using the evaluation results of certified PPs

While evaluating a PP that is based on one or more certified PPs, it may be possible to re-use the fact that these PPs were certified. The potential for re-use of the result of a certified PP is greater if the PP under evaluation does not add threats, OSPs, security objectives and/or security requirements to those of the PP that conformance is being claimed to. If the PP under evaluation contains much more than the certified PP, re-use may not be useful at all.

The evaluator is allowed to re-use the PP evaluation results by doing certain analyses only partially or not at all if these analyses or parts thereof were already done as part of the PP evaluation. While doing this, the evaluator should assume that the analyses in the PP were performed correctly.

An example would be where the PP that conformance is being claimed to contain a set of security requirements, and these were determined to be internally consistent during its evaluation. If the PP under evaluation uses the exact same requirements, the consistency analysis does not have to be repeated during the PP evaluation. If the PP under evaluation adds one or more requirements, or performs operations on these requirements, the analysis will have to be repeated. However, it may be possible to save work in this consistency analysis by using the fact that the original requirements are internally consistent. If the original requirements are internally consistent, the evaluator only has to determine that:

a) the set of all new and/or changed requirements is internally consistent, and

b) the set of all new and/or changed requirements is consistent with the original requirements.

The evaluator notes in the ETR each case where analyses are not done or only partially done for this reason.

8.3 PP introduction (APE_INT)

8.3.1 Evaluation of sub-activity (APE_INT.1)

8.3.1.1 Objectives

The objective of this sub-activity is to determine whether the PP is correctly identified, and whether the PP reference and TOE overview are consistent with each other.

8.3.1.2 Input

The evaluation evidence for this sub-activity is:

a) the PP.

8.3.1.3 Action APE_INT.1.1E

ISO/IEC 15408-3 APE_INT.1.1C: *The PP introduction shall contain a PP reference and a TOE overview.*

8.3.1.3.1 Work unit APE_INT.1-1

The evaluator **shall check** that the PP introduction contains a PP reference and a TOE overview.

ISO/IEC 15408-3 APE_INT.1.2C: *The PP reference shall uniquely identify the PP.*

1456 **8.3.1.3.2 Work unit APE_INT.1-2**

1457 The evaluator **shall examine** the PP reference to determine that it uniquely identifies the PP.

1458 The evaluator determines that the PP reference identifies the PP itself, so that it may be easily
1459 distinguished from other PPs, and that it also uniquely identifies each version of the PP, e.g. by
1460 including a version number and/or a date of publication.

1461 The PP should have some referencing system that is capable of supporting unique references (e.g.
1462 use of numbers, letters or dates).

1463 ISO/IEC 15408-3 APE_INT.1.3C: *The TOE overview shall summarise the usage and major security*
1464 *features of the TOE.*

1465 **8.3.1.3.3 Work unit APE_INT.1-3**

1466 The evaluator **shall examine** the TOE overview to determine that it describes the usage and major
1467 security features of the TOE.

1468 The TOE overview should briefly (i.e. several paragraphs) describe the usage and major security
1469 features expected of the TOE. The TOE overview should enable consumers and potential TOE
1470 developers to quickly determine whether the PP is of interest to them.

1471 The evaluator determines that the overview is clear enough for TOE developers and consumers,
1472 and sufficient to give them a general understanding of the intended usage and major security
1473 features of the TOE.

1474 ISO/IEC 15408-3 APE_INT.1.4C: *The TOE overview shall identify the TOE type.*

1475 **8.3.1.3.4 Work unit APE_INT.1-4**

1476 The evaluator **shall check** that the TOE overview identifies the TOE type.

1477 ISO/IEC 15408-3 APE_INT.1.5C: *The TOE overview shall identify any non-TOE*
1478 *hardware/software/firmware available to the TOE.*

1479 **8.3.1.3.5 Work unit APE_INT.1-5**

1480 The evaluator **shall examine** the TOE overview to determine that it identifies any non-TOE
1481 hardware/software/firmware available to the TOE.

1482 While some TOEs may run stand-alone, other TOEs (notably software TOEs) need additional
1483 hardware, software or firmware to operate. In this subclause of the PP, the PP author lists all
1484 hardware, software, and/or firmware that will be available for the TOE to run on.

1485 This identification should be detailed enough for potential consumers and TOE developers to
1486 determine whether their TOE may operate with the listed hardware, software and firmware.

1487 **8.4 Conformance claims (APE_CCL)**

1488 **8.4.1 Evaluation of sub-activity (APE_CCL.1)**

1489 **8.4.1.1 Objectives**

1490 The objective of this sub-activity is to determine the validity of various conformance claims. These
1491 describe how the PP conforms to ISO/IEC 15408, other PPs and packages.

1492 **8.4.1.2 Input**

1493 The evaluation evidence for this sub-activity is:

- 1494 a) the PP
- 1495 b) the content of the PP configuration
- 1496 c) the package(s) that the PP claims conformance to.

1497 **8.4.1.3 Action APE_CCL.1.1E**

1498 ISO/IEC 15408-3 APE_CCL.1.1C: *The conformance claim shall contain an ISO/IEC 15408*
1499 *conformance claim that identifies the version of ISO/IEC 15408 to which the PP claims conformance.*

1500 **8.4.1.3.1 Work unit APE_CCL.1-1**

1501 The evaluator **shall check** that the conformance claim contains an ISO/IEC 15408 conformance
1502 claim that identifies the version of ISO/IEC 15408 to which the PP claims conformance.

1503 The evaluator determines that ISO/IEC 15408 conformance claim identifies the version of ISO/IEC
1504 15408 that was used to develop this PP. This should include the version number of ISO/IEC 15408
1505 and, unless the International English version of ISO/IEC 15408 was used, the language of the
1506 version of ISO/IEC 15408 that was used.

1507 ISO/IEC 15408-3 APE_CCL.1.2C: *ISO/IEC 15408 conformance claim shall describe the conformance of*
1508 *the PP to ISO/IEC 15408-2 as either ISO/IEC 15408-2 conformant or ISO/IEC 15408-2 extended.*

1509 **8.4.1.3.2 Work unit APE_CCL.1-2**

1510 The evaluator **shall check** that ISO/IEC 15408 conformance claim states a claim of either ISO/IEC
1511 15408-2 conformant or ISO/IEC 15408-2 extended for the PP.

1512 ISO/IEC 15408-3 APE_CCL.1.3C: *ISO/IEC 15408 conformance claim shall describe the conformance of*
1513 *the PP to ISO/IEC 15408-3 as either ISO/IEC 15408-3 conformant or ISO/IEC 15408-3 extended.*

1514 **8.4.1.3.3 Work unit APE_CCL.1-3**

1515 The evaluator **shall check** that ISO/IEC 15408 conformance claim states a claim of either ISO/IEC
1516 15408-3 conformant or ISO/IEC 15408-3 extended for the PP.

1517 ISO/IEC 15408-3 APE_CCL.1.4C: *ISO/IEC 15408 conformance claim shall be consistent with the*
1518 *extended components definition.*

1519 **8.4.1.3.4 Work unit APE_CCL.1-4**

1520 The evaluator **shall examine** ISO/IEC 15408 conformance claim for ISO/IEC 15408-2 to determine
1521 that it is consistent with the extended components definition.

1522 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-2 conformant, the evaluator
1523 determines that the extended components definition does not define functional components.

1524 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-2 extended, the evaluator determines
1525 that the extended components definition defines at least one extended functional component.

1526 **8.4.1.3.5 Work unit APE_CCL.1-5**

1527 The evaluator **shall examine** ISO/IEC 15408 conformance claim for ISO/IEC 15408-3 to determine
1528 that it is consistent with the extended components definition.

1529 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-3 conformant, the evaluator
1530 determines that the extended components definition does not define assurance components.

1531 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-3 extended, the evaluator determines
1532 that the extended components definition defines at least one extended assurance component.

1533 ISO/IEC 15408-3 APE_CCL.1.5C: *The conformance claim shall identify all PPs and security*
1534 *requirement packages to which the PP claims conformance.*

1535 **8.4.1.3.6 Work unit APE_CCL.1-6**

1536 The evaluator **shall check** that the conformance claim contains a PP claim that identifies all PPs for
1537 which the PP claims conformance.

1538 If the PP does not claim conformance to another PP, this work unit is not applicable and therefore
1539 considered to be satisfied.

1540 The evaluator determines that any referenced PPs are unambiguously identified (e.g. by title and
1541 version number, or by the identification included in the introduction of that PP).

1542 If the PP claims conformance to another PP which uses mandatory functional packages, the
1543 evaluator determines that all mandatory packages from that PP are claimed and that they are
1544 unambiguously identified (e.g. by title and version number, or by the identification included in the
1545 introduction of that package).

1546 If the PP claims conformance to another PP which uses optional functional packages, the evaluator
1547 determines that any referenced optional packages from that PP are unambiguously identified (e.g.
1548 by title and version number, or by the identification included in the introduction of that package).

1549 If the PP claims conformance to another PP, which uses optional functional packages that have
1550 dependencies in their conformance, claim, the evaluator determines that all dependencies are met.

1551

1552 The evaluator is reminded that claims of partial conformance to a PP are not permitted.

1553 The evaluator **shall check** that, for each other PP to which the PP being evaluated claims
1554 conformance, the conformance statement of that other PP requires strict or demonstrable
1555 conformance.

1556 If the PP does not claim conformance to another PP, this work unit is not applicable and therefore
1557 considered to be satisfied.

1558 A PP is not allowed to claim conformance to a PP that requires exact conformance. As long as all
1559 PPs to which the PP is claiming conformance require only strict or demonstrable conformance, this
1560 work unit passes.

1561 **8.4.1.3.7 Work unit APE_CCL.1-7**

1562 The evaluator shall check, for each identified functional package, that the package definition is
1563 complete.

1564 If the PP does not claim conformance to a package, this work unit is not applicable and therefore
1565 considered to be satisfied.

1566 The evaluator determines that the package definition is conformant to the requirements from
1567 ISO/IEC 15408-1, clause 8 “Packages” by checking that the functional package includes:

- 1568 a) A functional package identification, giving a unique name, short name, version, date,
1569 sponsor, and the ISO/IEC 15408 edition;
- 1570 b) A functional package overview, giving a narrative description of the security functionality;
- 1571 c) A functional package conformance claim, giving the conformance claim to ISO/IEC 15408-
1572 2 and ISO/IEC 15408-3;
- 1573 d) If the package defines an SPD then it shall also either
 - 1574 i. if using the Direct Rationale approach: include a security functional
1575 requirements rationale that maps all threats, OSPs and assumptions in
1576 the SPD directly to the SFRs and Security Objectives for the operational
1577 environment; or else
 - 1578 ii. if not using the Direct Rationale approach: include Security Objectives for
1579 the TOE and the operational environment and the Security Objectives
1580 rationale;
- 1581 e) The functional package SFRs, and shall also include a security requirements rationale if the
1582 package includes any Security Objectives for the TOE.

1583

1584 ISO/IEC 15408-3 APE_CCL.1.6C: *The conformance claim shall describe any conformance of the PP to*
1585 *a package as either package-conformant or package-augmented.*

1586 8.4.1.3.8 Work unit APE_CCL.1-8

1587 The evaluator **shall check** that, for each identified package, the conformance claim states a claim of
1588 either package-name conformant or package-name augmented.

1589 If the PP does not claim conformance to a package, this work unit is not applicable and therefore
1590 considered to be satisfied.

1591 If the package conformance claim contains package-name conformant, the evaluator determines
1592 that:

- 1593 a) If the package is an assurance package, then the PP contains all SARs included in the
1594 package, but no additional SARs.
- 1595 b) If the package is a functional package, then all assumptions, threats, OSPs, security
1596 objectives and SFRs included in the package are included in identical form in the PP (after
1597 allowing for assignments and selections from the package to be completed as required by
1598 the PP).

1599 If the package conformance claim contains package-name augmented, the evaluator determines
1600 that:

- 1601 a) If the package is an assurance package, then the PP contains all SARs included in the package,
1602 and at least one additional SAR or at least one SAR that is hierarchical to a SAR in the package.
- 1603 b) If the package is a functional package, then all assumptions, threats, OPSs, Security Objectives,
1604 and SFRs included in the package are included in identical form in the PP (after allowing for
1605 assignments and selections from the package to be completed as required by the PP) except

- 1606 that the PP shall have at least one additional SFR or one SFR that is hierarchically higher than
1607 an SFR in the functional package.
- 1608 ISO/IEC 15408-3 APE_CCL.1.7C: *The conformance claim rationale shall demonstrate that the TOE*
1609 *type is consistent with the TOE type in the PPs for which conformance is being claimed.*
- 1610 **8.4.1.3.9 Work unit APE_CCL.1-9**
- 1611 The evaluator ***shall examine*** the conformance claim rationale to determine that the TOE type of
1612 the TOE is consistent with all TOE types of the PPs.
- 1613 If the PP does not claim conformance to another PP, this work unit is not applicable and therefore
1614 considered to be satisfied.
- 1615 The relation between the types may be simple: a firewall PP claiming conformance to another
1616 firewall PP, or more complex: a smart card PP claiming conformance to a number of other PPs at
1617 the same time: a PP for the integrated circuit, a PP for the smart card OS, and two PPs for two
1618 applications on the smart card.
- 1619 ISO/IEC 15408-3 APE_CCL.1.8C: *The conformance claim rationale shall demonstrate that the*
1620 *statement of the security problem definition is consistent with the statement of the security problem*
1621 *definition in the PPs for which conformance is being claimed.*
- 1622 **8.4.1.3.10 Work unit APE_CCL.1-10**
- 1623 The evaluator ***shall examine*** the conformance claim rationale to determine that it demonstrates
1624 that the statement of security problem definition is consistent, as defined by the conformance
1625 statement of the PP, with the statements of security problem definition stated in the PPs to which
1626 conformance is being claimed.
- 1627 If the PP under evaluation does not claim conformance with another PP, this work unit is not
1628 applicable and therefore considered to be satisfied.
- 1629 If the PP to which conformance is being claimed does not have a statement of security problem
1630 definition, this work unit is not applicable and therefore considered to be satisfied.
- 1631 If the PP to which conformance is being claimed contains functional packages, the evaluator
1632 determines that the security problem definition of the PP under evaluation consists of all
1633 assumptions, threats and OSPs of all mandatory and all selected optional functional packages.
- 1634 The terms exact, strict and demonstrable conformance are defined in ISO/IEC 15408 Part 1.
- 1635 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
1636 demonstrable conformance also hold for the SPD descriptions taken from the packages.
- 1637 Note that since a PP can only claim conformance to another PP whose conformance statement
1638 requires strict or demonstrable conformance, those are the only cases covered in the following
1639 paragraphs. If strict conformance is required by the PP to which conformance is being claimed, no
1640 conformance claim rationale is required. Instead, the evaluator determines whether:
- 1641 a) the threats in the PP under evaluation are a superset of or identical to the threats in the
1642 PP to which conformance is being claimed;
- 1643 b) the OSPs in the PP under evaluation are a superset of or identical to the OSPs in the PP to
1644 which conformance is being claimed;

- 1645 c) the assumptions in the PP claiming conformance are identical to the assumptions in the
1646 PP to which conformance is being claimed, with two possible exceptions described in the
1647 following two bullet points;
- 1648 — an assumption (or part of an assumption) from the PP to which conformance is claimed, can be
1649 omitted, if all security objectives for the operational environment addressing this assumption
1650 (or part of an assumption) are replaced by security objectives for the TOE;
- 1651 — an assumption can be added to the assumptions defined in the PP to which conformance is
1652 claimed, if a justification is given, why the new assumption neither mitigates a threat (or a part
1653 of a threat) meant to be addressed by security objectives for the TOE in the PP to which
1654 conformance is claimed, nor fulfils an OSP (or part of an OSP) meant to be addressed by
1655 security objectives for the TOE in the PP to which conformance is claimed.
- 1656 When examining a PP, which omits assumptions from another PP to which conformance is claimed,
1657 or adds new assumptions, the evaluator shall carefully determine, if the conditions given above are
1658 fulfilled. The following discussion gives some motivation and examples for these cases:
- 1659 — Example for omitting an assumption: A PP to which conformance is claimed, may contain an
1660 assumption stating that the operational environment prevents unauthorized modification or
1661 interception of data sent to an external interface of the TOE. This may be the case if the TOE
1662 accepts data in clear text and without integrity protection at this interface and is assumed to be
1663 located in a secure operational environment, which will prevent attackers from accessing these
1664 data. The assumption will then be mapped in the PP, to which conformance is claimed, to some
1665 objective for the operational environment stating that the data interchanged at this interface
1666 are protected by adequate measures in the operational environment. If a PP claiming this PP,
1667 defines a more secure TOE, which has an additional security objective stating that the TOE
1668 itself protects these data, for example by providing a secure channel for encryption and
1669 integrity protection of all data transferred via this interface, the corresponding objective and
1670 assumption for the operational environment can be omitted from the PP claiming conformance.
1671 This is also called re-assigning of the objective, since the objective is re-assigned from the
1672 operational environment to the TOE. Note, that this TOE is still secure in an operational
1673 environment fulfilling the omitted assumption and therefore still fulfils the PP to which
1674 conformance is claimed.
- 1675 — Example for adding an assumption: In this example, the PP to which conformance is claimed, is
1676 designed to specify requirements for a TOE of type "Firewall" and the author of another PP
1677 wishes to claim conformance to this PP for a TOE, which implements a firewall, but
1678 additionally provides the functionality of a virtual private network (VPN) component. For the
1679 VPN functionality, the TOE needs cryptographic keys and these keys may also have to be
1680 handled securely by the operational environment (e. g. if symmetric keys are used to secure
1681 the network connection and therefore need to be provided in some secure way to other
1682 components in the network). In this case, it is acceptable to add an assumption that the
1683 cryptographic keys used by the VPN are handled securely by the operational environment.
1684 This assumption does not address threats or OSPs of the PP to which conformance is claimed,
1685 and therefore fulfils the conditions stated above.
- 1686 — Counterexample for adding an assumption: In a variant of the first example a PP to which
1687 conformance is claimed, may already contain an objective for the TOE to provide a secure
1688 channel for one of its interfaces, and this objective is mapped to a threat of unauthorized
1689 modification or reading of the data on this interface. In this case, it is clearly not allowed for
1690 another PP claiming this PP, to add an assumption for the operational environment, which
1691 assumes that the operational environment protects data on this interface against modification
1692 or unauthorized reading of the data. This assumption would reduce a threat, which is meant to
1693 be addressed by the TOE. Therefore, a TOE fulfilling a PP with this added assumption would
1694 not automatically fulfil the PP to which conformance is claimed, anymore and this addition is
1695 therefore not allowed.

1696 — Second counterexample for adding an assumption: In the example above of a TOE
 1697 implementing a firewall it would not be admissible to add a general assumption that the TOE is
 1698 only connected to trusted devices, because this would obviously remove essential threats
 1699 relevant for a firewall (namely that there is untrusted IP traffic, which needs to be filtered).
 1700 Therefore, this addition would not be allowed.

1701 If demonstrable conformance is required by the PP to which conformance is being claimed, the
 1702 evaluator examines the conformance claim rationale to determine that it demonstrates that the
 1703 statement of security problem definition of the PP under evaluation is equivalent or more
 1704 restrictive than the statement of security problem definition in the PP to which conformance is
 1705 being claimed.

1706 For this, the conformance claim rationale needs to demonstrate that the security problem
 1707 definition in the PP claiming conformance is equivalent (or more restrictive) than the security
 1708 problem definition in the PP to which conformance is claimed. This means that:

1709 — all TOEs that would meet the security problem definition in the PP claiming conformance also
 1710 meet the security problem definition in the PP to which conformance is claimed. This can also
 1711 be shown indirectly by demonstrating that every event, which realizes a threat defined in the
 1712 PP to which conformance is claimed, or violates an OSP defined in the PP to which
 1713 conformance is claimed, would also realize a threat stated in the PP claiming conformance or
 1714 violate an OSP defined in the PP claiming conformance. Note that fulfilling an OSP stated in the
 1715 PP claiming conformance may avert a threat stated in the PP to which conformance is claimed,
 1716 or that averting a threat stated in the PP claiming conformance may fulfil an OSP stated in the
 1717 PP to which conformance is claimed, so threats and OSPs can substitute each other;

1718 — all operational environments that would meet the security problem definition in the PP to
 1719 which conformance is claimed, would also meet the security problem definition in the PP
 1720 claiming conformance (with one exception in the next bullet);

1721 — besides a set of assumptions in the PP claiming conformance needed to demonstrate
 1722 conformance to the SPD of the PP to which conformance is claimed, an PP claiming
 1723 conformance may specify further assumptions, but only if these additional assumptions are
 1724 independent of and do not affect the security problem definition as defined in the PP to which
 1725 conformance is claimed. More detailed, there are no assumptions in the PP claiming
 1726 conformance that exclude threats to the TOE that need to be countered by the TOE according
 1727 to the PP to which conformance is claimed. Similarly, there are no assumptions in the PP
 1728 claiming conformance that realize aspects of an OSP stated in the PP to which conformance is
 1729 claimed, which are meant to be fulfilled by the TOE according to the PP to which conformance
 1730 is claimed.

1731 ISO/IEC 15408-3 APE_CCL.1.9C: *The conformance claim rationale shall demonstrate that the*
 1732 *statement of security objectives is consistent with the statement of security objectives in the PPs for*
 1733 *which conformance is being claimed.*

1734 **8.4.1.3.11 Work unit APE_CCL.1-11**

1735 The evaluator **shall examine** the conformance claim rationale to determine that the statement of
 1736 security objectives is consistent, as defined by the conformance statement of the PPs, with the
 1737 statement of security objectives in the PPs.

1738 If the PP does not claim conformance to another PP, this work unit is not applicable and therefore
 1739 considered to be satisfied.

1740 If the PP to which conformance is being claimed contains functional packages, the evaluator
 1741 determines that the security objectives of the PP under evaluation consist of all security objectives
 1742 of all mandatory and all selected optional functional packages.

1743 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
1744 demonstrable conformance also hold for the security objectives taken from the packages.

1745 Note that since a PP can only claim conformance to another PP whose conformance statement
1746 requires strict or demonstrable conformance, those are the only cases covered in the following
1747 paragraphs. If strict conformance is required by the PP to which conformance is being claimed, no
1748 conformance claim rationale is required. Instead, the evaluator determines whether:

1749 — The PP under evaluation contains all security objectives for the TOE of the PP to which
1750 conformance is being claimed. Note that it is allowed for the PP under evaluation to have
1751 additional security objectives for the TOE;

1752 — The security objectives for the operational environment in the PP claiming conformance are
1753 identical to the security objectives for the operational environment in the PP to which
1754 conformance is being claimed, with two possible exceptions described in the following two
1755 bullet points;

1756 — a security objective for the operational environment (or part of such security objective) from
1757 the PP to which conformance is claimed, can be replaced by the same (part of the) security
1758 objective stated for the TOE;

1759 — a security objective for the operational environment can be added to the objectives defined in
1760 the PP to which conformance is claimed, if a justification is given, why the new objective
1761 neither mitigates a threat (or a part of a threat) meant to be addressed by security objectives
1762 for the TOE in the PP to which conformance is claimed, nor fulfils an OSP (or part of an OSP)
1763 meant to be addressed by security objectives for the TOE in the PP to which conformance is
1764 claimed.

1765 When examining a PP claiming another PP which omits security objectives for the operational
1766 environment from the PP to which conformance is claimed, or adds new security objectives for the
1767 operational environment, the evaluator shall carefully determine, if the conditions given above are
1768 fulfilled. The examples given for the case of assumptions in the preceding work unit are also valid
1769 here.

1770 If demonstrable conformance is required by the PP to which conformance is being claimed, the
1771 evaluator examines the conformance claim rationale to determine that it demonstrates that the
1772 statement of security objectives of the PP under evaluation is equivalent or more restrictive than
1773 the statement of security objectives in the PP to which conformance is being claimed.

1774 For this the conformance claim rationale needs to demonstrate that the security objectives in the
1775 PP claiming conformance are equivalent (or more restrictive) than the security objectives in the PP
1776 to which conformance is claimed. This means that:

1777 — all TOEs that would meet the security objectives for the TOE in the PP claiming conformance
1778 also meet the security objectives for the TOE in the PP to which conformance is claimed;

1779 — all operational environments that would meet the security objectives for the operational
1780 environment in the PP to which conformance is claimed, would also meet the security
1781 objectives for the operational environment in the PP claiming conformance (with one
1782 exception in the next bullet);

1783 — besides a set of security objectives for the operational environment in the PP claiming
1784 conformance, which are used to demonstrate conformance to the set of security objectives
1785 defined in the PP to which conformance is claimed, an PP claiming conformance may specify
1786 further security objectives for the operational environment, but only if these security
1787 objectives neither affect the original set of security objectives for the TOE nor the security

- 1788 objectives for the operational environment as defined in the PP to which conformance is
1789 claimed.
- 1790 ISO/IEC 15408-3 APE_CCL.1.10C: *The conformance claim rationale shall demonstrate that the*
1791 *statement of security requirements is consistent with the statement of security requirements in the*
1792 *PPs for which conformance is being claimed.*
- 1793 **8.4.1.3.12 Work unit APE_CCL.1-12**
- 1794 The evaluator **shall examine** the PP to determine that it is consistent, as defined by the
1795 conformance statement of the PP, with all security requirements in the PPs for which conformance
1796 is being claimed.
- 1797 If the PP does not claim conformance to another PP, this work unit is not applicable and therefore
1798 considered to be satisfied.
- 1799 If the PP to which conformance is being claimed contains functional packages, the evaluator
1800 determines that the SFRs of the PP under evaluation consist of all SFRs (or hierarchical SFRs) of all
1801 mandatory and all selected optional functional packages.
- 1802 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
1803 demonstrable conformance also hold for the SFRs taken from the packages.
- 1804 — Note that since a PP can only claim conformance to another PP whose conformance statement
1805 requires strict or demonstrable conformance, those are the only cases covered in the following
1806 paragraphs.
- 1807 If strict conformance is required by the PP to which conformance is being claimed, no conformance
1808 claim rationale is required. Instead, the evaluator determines whether the statement of security
1809 requirements in the PP under evaluation is a superset of or identical to the statement of security
1810 requirements in the PP to which conformance is being claimed (for strict conformance).
- 1811 If demonstrable conformance is required by the PP to which conformance is being claimed, the
1812 evaluator examines the conformance claim rationale to determine that it demonstrates that the
1813 statement of security requirements of the PP under evaluation is equivalent or more restrictive
1814 than the statement of security requirements in the PP to which conformance is being claimed.
- 1815 For:
- 1816 — SFRs: The conformance rationale in the PP claiming conformance shall demonstrate that the
1817 overall set of requirements defined by the SFRs in the PP claiming conformance is equivalent
1818 (or more restrictive) than the overall set of requirements defined by the SFRs in the PP to
1819 which conformance is claimed. This means that all TOEs that would meet the requirements
1820 defined by the set of all SFRs in the PP claiming conformance would also meet the
1821 requirements defined by the set of all SFRs in the PP to which conformance is claimed;
- 1822 — SARs: The PP claiming conformance shall contain all SARs in the PP to which conformance is
1823 claimed, but may claim additional SARs or replace SARs by hierarchically stronger SARs. The
1824 completion of operations in the PP claiming conformance must be consistent with that in the
1825 PP to which conformance is claimed; either the same completion will be used in the PP
1826 claiming conformance as that in the PP to which conformance is claimed or a completion that
1827 makes the SAR more restrictive (the rules of refinement apply).
- 1828 ISO/IEC 15408-3 APE_CCL.1.11C: *The conformance statement shall describe the conformance*
1829 *required of any PPs/STs to the PP as exact-PP, strict-PP or demonstrable-PP conformance.*

1830 **8.4.1.3.13 Work unit APE_CCL.1-13**

1831 The evaluator **shall check** that the PP conformance statement states a claim of exact-PP, strict-PP
1832 or demonstrable-PP conformance.

1833 ISO/IEC 15408-3 APE_CCL.1.12C: *The conformance statement shall identify the set of other PPs (if*
1834 *any) to which, in combination with the PP under evaluation, exact conformance is allowed to be*
1835 *claimed.*

1836 **8.4.1.3.14 Work unit APE_CCL.1-14**

1837 The evaluator **shall check** the conformance statement to determine that it lists the set of PPs to
1838 which, in combination with the PP being evaluated, an exact conformance claim (in an ST or PP
1839 Configuration) is allowed.

1840 If the PP does not require exact conformance in its conformance statement, this work unit does not
1841 apply and is therefore considered satisfied.

1842 If the PP does not allow claims of exact conformance to it in combination with any other PPs, then
1843 no list of PPs is required and this work unit is considered satisfied.

1844 There are no other actions for the evaluator other than determining that the list is present.

1845 **8.4.1.3.15 Work unit APE_CCL.1-15**

1846 ***[**This work unit has been deleted and renumbering of later work units may therefore be***
1847 ***done in a future draft]***

1848 ISO/IEC 15408-3 APE_CCL.1.13C: *The conformance statement shall identify the set of PP-modules (if*
1849 *any) that are allowed to be used with the PP under evaluation in a PP-Configuration.*

1850 **8.4.1.3.16 Work unit APE_CCL.1-16**

1851 The evaluator shall check the conformance statement to determine that it lists the set of PP-
1852 Modules that can be used with the PP under evaluation in a PP-configuration.

1853 If the PP does not require exact conformance in its conformance statement, this work unit does not
1854 apply and is therefore considered satisfied.

1855 If the PP is not allowed to be used in a PP-Configuration, then the evaluator confirms that no PP-
1856 modules are listed.

1857 There are no other actions for the evaluator other than determining that the list is present.

1858 **8.5 Security problem definition (APE_SPD)**

1859 **8.5.1 Evaluation of sub-activity (APE_SPD.1)**

1860 **8.5.1.1 Objectives**

1861 The objective of this sub-activity is to determine that the security problem intended to be
1862 addressed by the TOE and its operational environment is clearly defined.

1863 **8.5.1.2 Input**

1864 The evaluation evidence for this sub-activity is:

1865 a) the PP.

1866	8.5.1.3 Action APE_SPD.1.1E
1867	ISO/IEC 15408-3 APE_SPD.1.1C: <i>The security problem definition shall describe the threats.</i>
1868	8.5.1.3.1 Work unit APE_SPD.1-1
1869	The evaluator <i>shall check</i> that the security problem definition describes the threats.
1870	If all security objectives are derived from assumptions and/or OSPs only, the statement of threats
1871	need not be present in the PP. In this case, this work unit is not applicable and therefore considered
1872	to be satisfied.
1873	The evaluator determines that the security problem definition describes the threats that must be
1874	countered by the TOE and/or its operational environment.
1875	ISO/IEC 15408-3 APE_SPD.1.2C: <i>All threats shall be described in terms of a threat agent, an asset,</i>
1876	<i>and an adverse action.</i>
1877	8.5.1.3.2 Work unit APE_SPD.1-2
1878	The evaluator <i>shall examine</i> the security problem definition to determine that all threats are
1879	described in terms of a threat agent, an asset, and an adverse action.
1880	If all security objectives are derived from assumptions and OSPs only, the statement of threats
1881	need not be present in the PP. In this case, this work unit is not applicable and therefore considered
1882	to be satisfied.
1883	Threat agents may be further described by aspects such as expertise, resource, opportunity, and
1884	motivation.
1885	ISO/IEC 15408-3 APE_SPD.1.3C: <i>The security problem definition shall describe the OSPs.</i>
1886	8.5.1.3.3 Work unit APE_SPD.1-3
1887	The evaluator <i>shall examine</i> that the security problem definition describes the OSPs.
1888	If all security objectives are derived from assumptions and/or threats only, OSPs need not be
1889	present in the PP. In this case, this work unit is not applicable and therefore considered to be
1890	satisfied.
1891	The evaluator determines that OSP statements are made in terms of rules or guidelines that must
1892	be followed by the TOE and/or its operational environment.
1893	The evaluator determines that each OSP is explained and/or interpreted in sufficient detail to make
1894	it clearly understandable; a clear presentation of policy statements is necessary to permit tracing
1895	security objectives to them.
1896	ISO/IEC 15408-3 APE_SPD.1.4C: <i>The security problem definition shall describe the assumptions</i>
1897	<i>about the operational environment of the TOE.</i>
1898	8.5.1.3.4 Work unit APE_SPD.1-4
1899	The evaluator <i>shall examine</i> the security problem definition to determine that it describes the
1900	assumptions about the operational environment of the TOE.
1901	If there are no assumptions, this work unit is not applicable and is therefore considered to be
1902	satisfied.

1903 The evaluator determines that each assumption about the operational environment of the TOE is
1904 explained in sufficient detail to enable consumers to determine that their operational environment
1905 matches the assumption. If the assumptions are not clearly understood, the end result may be that
1906 the TOE is used in an operational environment in which it will not function in a secure manner.

1907 **8.6 Security objectives (APE_OBJ)**

1908 **8.6.1 Evaluation of sub-activity (APE_OBJ.1)**

1909 **8.6.1.1 Objectives**

1910 The objective of this sub-activity is to determine whether the security objectives for the
1911 operational environment are clearly defined.

1912 **8.6.1.2 Input**

1913 The evaluation evidence for this sub-activity is:

1914 a) the PP.

1915 **8.6.1.3 Action APE_OBJ.1.1E**

1916 ISO/IEC 15408-3 APE_OBJ.1.1C: *The statement of security objectives shall describe the security*
1917 *objectives for the operational environment.*

1918 **8.6.1.3.1 Work unit APE_OBJ.1-1**

1919 The evaluator ***shall check*** that the statement of security objectives defines the security objectives
1920 for the operational environment.

1921 The evaluator checks that the security objectives for the operational environment are identified.

1922 **8.6.2 Evaluation of sub-activity (APE_OBJ.2)**

1923 **8.6.2.1 Objectives**

1924 The objective of this sub-activity is to determine whether the security objectives adequately and
1925 completely address the security problem definition and that the division of this problem between
1926 the TOE and its operational environment is clearly defined.

1927 **8.6.2.2 Input**

1928 The evaluation evidence for this sub-activity is:

1929 a) the PP.

1930 **8.6.2.3 Action APE_OBJ.2.1E**

1931 ISO/IEC 15408-3 APE_OBJ.2.1C: *The statement of security objectives shall describe the security*
1932 *objectives for the TOE and the security objectives for the operational environment.*

1933 **8.6.2.3.1 Work unit APE_OBJ.2-1**

1934 The evaluator ***shall check*** that the statement of security objectives defines the security objectives
1935 for the TOE and the security objectives for the operational environment.

1936 The evaluator checks that both categories of security objectives are clearly identified and
1937 separated from the other category.

1938	ISO/IEC 15408-3 APE_OBJ.2.2C: <i>The security objectives rationale shall trace each security objective</i>
1939	<i>for the TOE back to threats countered by that security objective and OSPs enforced by that security</i>
1940	<i>objective.</i>
1941	8.6.2.3.2 Work unit APE_OBJ.2-2
1942	The evaluator <i>shall check</i> that the security objectives rationale traces all security objectives for the
1943	TOE back to threats countered by the objectives and/or OSPs enforced by the objectives.
1944	Each security objective for the TOE may trace back to threats or OSPs, or a combination of threats
1945	and OSPs, but it must trace back to at least one threat or OSP.
1946	Failure to trace implies that either the security objectives rationale is incomplete, the security
1947	problem definition is incomplete, or the security objective for the TOE has no useful purpose.
1948	ISO/IEC 15408-3 APE_OBJ.2.3C: <i>The security objectives rationale shall trace each security objective</i>
1949	<i>for the operational environment back to threats countered by that security objective, OSPs enforced</i>
1950	<i>by that security objective, and assumptions upheld by that security objective.</i>
1951	8.6.2.3.3 Work unit APE_OBJ.2-3
1952	The evaluator <i>shall check</i> that the security objectives rationale traces the security objectives for
1953	the operational environment back to threats countered by that security objective, to OSPs enforced
1954	by that security objective, and to assumptions upheld by that security objective.
1955	Each security objective for the operational environment may trace back to threats, OSPs,
1956	assumptions, or a combination of threats, OSPs and/or assumptions, but it must trace back to at
1957	least one threat, OSP or assumption.
1958	Failure to trace implies that either the security objectives rationale is incomplete, the security
1959	problem definition is incomplete, or the security objective for the operational environment has no
1960	useful purpose.
1961	ISO/IEC 15408-3 APE_OBJ.2.4C: <i>The security objectives rationale shall demonstrate that the security</i>
1962	<i>objectives counter all threats.</i>
1963	8.6.2.3.4 Work unit APE_OBJ.2-4
1964	The evaluator <i>shall examine</i> the security objectives rationale to determine that it justifies for each
1965	threat that the security objectives are suitable to counter that threat.
1966	If no security objectives trace back to the threat, the evaluator action related to this work unit is
1967	assigned a fail verdict.
1968	The evaluator determines that the justification for a threat shows whether the threat is removed,
1969	diminished or mitigated.
1970	The evaluator determines that the justification for a threat demonstrates that the security
1971	objectives are sufficient: if all security objectives that trace back to the threat are achieved, the
1972	threat is removed, sufficiently diminished, or the effects of the threat are sufficiently mitigated.
1973	Note that the tracings from security objectives to threats provided in the security objectives
1974	rationale may be part of a justification, but do not constitute a justification by themselves. Even in
1975	the case that a security objective is merely a statement reflecting the intent to prevent a particular
1976	threat from being realised, a justification is required, but this justification may be as minimal as
1977	"Security Objective X directly counters Threat Y".

1978	The evaluator also determines that each security objective that traces back to a threat is necessary:
1979	when the security objective is achieved it actually contributes to the removal, diminishing or
1980	mitigation of that threat.
1981	ISO/IEC 15408-3 APE_OBJ.2.5C: <i>The security objectives rationale shall demonstrate that the security</i>
1982	<i>objectives enforce all OSPs.</i>
1983	8.6.2.3.5 Work unit APE_OBJ.2-5
1984	The evaluator shall examine the security objectives rationale to determine that for each OSP it
1985	justifies that the security objectives are suitable to enforce that OSP.
1986	If no security objectives trace back to the OSP, the evaluator action related to this work unit is
1987	assigned a fail verdict.
1988	The evaluator determines that the justification for an OSP demonstrates that the security
1989	objectives are sufficient: if all security objectives that trace back to that OSP are achieved, the OSP
1990	is enforced.
1991	The evaluator also determines that each security objective that traces back to an OSP is necessary:
1992	when the security objective is achieved it actually contributes to the enforcement of the OSP.
1993	Note that the tracings from security objectives to OSPs provided in the security objectives rationale
1994	may be part of a justification, but do not constitute a justification by themselves. In the case that a
1995	security objective is merely a statement reflecting the intent to enforce a particular OSP, a
1996	justification is required, but this justification may be as minimal as "Security Objective X directly
1997	enforces OSP Y".
1998	ISO/IEC 15408-3 APE_OBJ.2.6C: <i>The security objectives rationale shall demonstrate that the security</i>
1999	<i>objectives for the operational environment uphold all assumptions.</i>
2000	8.6.2.3.6 Work unit APE_OBJ.2-6
2001	The evaluator shall examine the security objectives rationale to determine that for each
2002	assumption for the operational environment it contains an appropriate justification that the
2003	security objectives for the operational environment are suitable to uphold that assumption.
2004	If no security objectives for the operational environment trace back to the assumption, the
2005	evaluator action related to this work unit is assigned a fail verdict.
2006	The evaluator determines that the justification for an assumption about the operational
2007	environment of the TOE demonstrates that the security objectives are sufficient: if all security
2008	objectives for the operational environment that trace back to that assumption are achieved, the
2009	operational environment upholds the assumption.
2010	The evaluator also determines that each security objective for the operational environment that
2011	traces back to an assumption about the operational environment of the TOE is necessary: when the
2012	security objective is achieved it actually contributes to the operational environment upholding the
2013	assumption.
2014	Note that the tracings from security objectives for the operational environment to assumptions
2015	provided in the security objectives rationale may be a part of a justification, but do not constitute a
2016	justification by themselves. Even in the case that a security objective of the operational
2017	environment is merely a restatement of an assumption, a justification is required, but this
2018	justification may be as minimal as "Security Objective X directly upholds Assumption Y".

2019	8.7 Extended components definition (APE_ECD)
2020	8.7.1 Evaluation of sub-activity (APE_ECD.1)
2021	8.7.1.1 Objectives
2022	The objective of this sub-activity is to determine whether extended components have been clearly and unambiguously defined, and whether they are necessary, i.e. they may not be clearly expressed using existing ISO/IEC 15408-2 or ISO/IEC 15408-3 components.
2023	
2024	
2025	8.7.1.2 Input
2026	The evaluation evidence for this sub-activity is:
2027	a) the PP.
2028	8.7.1.3 Action APE_ECD.1.1E
2029	ISO/IEC 15408-3 APE_ECD.1.1C: <i>The statement of security requirements shall identify all extended security requirements.</i>
2030	
2031	8.7.1.3.1 Work unit APE_ECD.1-1
2032	The evaluator shall check that all security requirements in the statement of security requirements that are not identified as extended requirements are present in ISO/IEC 15408-2 or in ISO/IEC 15408-3.
2033	
2034	
2035	ISO/IEC 15408-3 APE_ECD.1.2C: <i>The extended components definition shall define an extended component for each extended security requirement.</i>
2036	
2037	8.7.1.3.2 Work unit APE_ECD.1-2
2038	The evaluator shall check that the extended components definition defines an extended component for each extended security requirement.
2039	
2040	If the PP does not contain extended security requirements, this work unit is not applicable and therefore considered to be satisfied.
2041	
2042	A single extended component may be used to define multiple iterations of an extended security requirement, it is not necessary to repeat this definition for each iteration.
2043	
2044	ISO/IEC 15408-3 APE_ECD.1.3C: <i>The extended components definition shall describe how each extended component is related to the existing ISO/IEC 15408 components, families, and classes.</i>
2045	
2046	8.7.1.3.3 Work unit APE_ECD.1-3
2047	The evaluator shall examine the extended components definition to determine that it describes how each extended component fits into the existing ISO/IEC 15408 components, families, and classes.
2048	
2049	
2050	If the PP does not contain extended security requirements, this work unit is not applicable and therefore considered to be satisfied.
2051	
2052	The evaluator determines that each extended component is either:
2053	a) a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 family, or
2054	b) a member of a new family defined in the PP.

2055 If the extended component is a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 family,
2056 the evaluator determines that the extended components definition adequately describes why the
2057 extended component should be a member of that family and how it relates to other components of
2058 that family.

2059 If the extended component is a member of a new family defined in the PP, the evaluator confirms
2060 that the extended component is not appropriate for an existing family.

2061 If the PP defines new families, the evaluator determines that each new family is either:

2062 a) a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 class, or

2063 a member of a new class defined in the PP.

2064 If the family is a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 class, the evaluator
2065 determines that the extended components definition adequately describes why the family should
2066 be a member of that class and how it relates to other families in that class.

2067 If the family is a member of a new class defined in the PP, the evaluator confirms that the family is
2068 not appropriate for an existing class.

2069 **8.7.1.3.4 Work unit APE_ECD.1-4**

2070 The evaluator **shall examine** the extended components definition to determine that each definition
2071 of an extended component identifies all applicable dependencies of that component.

2072 If the PP does not contain extended security requirements, this work unit is not applicable and
2073 therefore considered to be satisfied.

2074 The evaluator confirms that no applicable dependencies have been overlooked by the PP author.

2075 ISO/IEC 15408-3 APE_ECD.1.4C: *The extended components definition shall use the existing ISO/IEC*
2076 *15408 components, families, classes, and methodology as a model for presentation.*

2077 **8.7.1.3.5 Work unit APE_ECD.1-5**

2078 The evaluator **shall examine** the extended components definition to determine that each extended
2079 functional component uses the existing ISO/IEC 15408-2 components as a model for presentation.

2080 If the PP does not contain extended SFRs, this work unit is not applicable and therefore considered
2081 to be satisfied.

2082 The evaluator determines that the extended functional component is consistent with ISO/IEC
2083 15408-2 Subclause **6.1.3, Component structure**.

2084 If the extended functional component uses operations, the evaluator determines that the extended
2085 functional component is consistent with ISO/IEC 15408-1 Subclause **7.1, Operations**.

2086 If the extended functional component is hierarchical to an existing functional component, the
2087 evaluator determines that the extended functional component is consistent with ISO/IEC 15408-2
2088 Subclause **6.2.1, Component changes highlighting**.

2089 **8.7.1.3.6 Work unit APE_ECD.1-6**

2090 The evaluator **shall examine** the extended components definition to determine that each definition
2091 of a new functional family uses the existing ISO/IEC 15408 functional families as a model for
2092 presentation.

- 2093 If the PP does not define new functional families, this work unit is not applicable and therefore
2094 considered to be satisfied.
- 2095 The evaluator determines that all new functional families are defined consistent with ISO/IEC
2096 15408-2 Subclause 6.1.2, **Family structure**.
- 2097 **8.7.1.3.7 Work unit APE_ECD.1-7**
- 2098 The evaluator *shall examine* the extended components definition to determine that each definition
2099 of a new functional class uses the existing ISO/IEC 15408 functional classes as a model for
2100 presentation.
- 2101 If the PP does not define new functional classes, this work unit is not applicable and therefore
2102 considered to be satisfied.
- 2103 The evaluator determines that all new functional classes are defined consistent with ISO/IEC
2104 15408-2 Subclause 6.1.1, **Class structure**.
- 2105 **8.7.1.3.8 Work unit APE_ECD.1-8**
- 2106 The evaluator *shall examine* the extended components definition to determine that each definition
2107 of an extended assurance component uses the existing ISO/IEC 15408-3 components as a model
2108 for presentation.
- 2109 If the PP does not contain extended SARs, this work unit is not applicable and therefore considered
2110 to be satisfied.
- 2111 The evaluator determines that the extended assurance component definition is consistent with
2112 ISO/IEC 15408-3 Subclause 6.1.3, **Assurance component structure**.
- 2113 If the extended assurance component uses operations, the evaluator determines that the extended
2114 assurance component is consistent with ISO/IEC 15408-1 Subclause 7.1, **Operations**.
- 2115 If the extended assurance component is hierarchical to an existing assurance component, the
2116 evaluator determines that the extended assurance component is consistent with ISO/IEC 15408-3
2117 Subclause 6.1.3, **Assurance component structure**.
- 2118 **8.7.1.3.9 Work unit APE_ECD.1-9**
- 2119 The evaluator *shall examine* the extended components definition to determine that, for each
2120 defined extended assurance component, applicable methodology has been provided.
- 2121 If the PP does not contain extended SARs, this work unit is not applicable and therefore considered
2122 to be satisfied.
- 2123 The evaluator determines that, for each evaluator action element of each extended SAR, one or
2124 more work units are provided and that successfully performing all work units for a given evaluator
2125 action element will demonstrate that the element has been achieved.
- 2126 **8.7.1.3.10 Work unit APE_ECD.1-10**
- 2127 The evaluator *shall examine* the extended components definition to determine that each definition
2128 of a new assurance family uses the existing ISO/IEC 15408 assurance families as a model for
2129 presentation.
- 2130 If the PP does not define new assurance families, this work unit is not applicable and therefore
2131 considered to be satisfied.

2132 The evaluator determines that all new assurance families are defined consistent with ISO/IEC
2133 15408-3 Subclause 6.1.2, Assurance family structure.

2134 **8.7.1.3.11 Work unit APE_ECD.1-11**

2135 The evaluator *shall examine* the extended components definition to determine that each definition
2136 of a new assurance class uses the existing ISO/IEC 15408 assurance classes as a model for
2137 presentation.

2138 If the PP does not define new assurance classes, this work unit is not applicable and therefore
2139 considered to be satisfied.

2140 The evaluator determines that all new assurance classes are defined consistent with ISO/IEC
2141 15408-3 Subclause 6.1.1, Assurance class structure.

2142 ISO/IEC 15408-3 APE_ECD.1.5C: *The extended components shall consist of measurable and objective*
2143 *elements such that conformance or nonconformance to these elements can be demonstrated.*

2144 **8.7.1.3.12 Work unit APE_ECD.1-12**

2145 The evaluator *shall examine* the extended components definition to determine that each element
2146 in each extended component is measurable and states objective evaluation requirements, such that
2147 conformance or nonconformance can be demonstrated.

2148 If the PP does not contain extended security requirements, this work unit is not applicable and
2149 therefore considered to be satisfied.

2150 The evaluator determines that elements of extended functional components are stated in such a
2151 way that they are testable, and traceable through the appropriate TSF representations.

2152 The evaluator also determines that elements of extended assurance components avoid the need for
2153 subjective evaluator judgement.

2154 The evaluator is reminded that whilst being measurable and objective is appropriate for all
2155 evaluation criteria, it is acknowledged that no formal method exists to prove such properties.
2156 Therefore the existing ISO/IEC 15408 functional and assurance components are to be used as a
2157 model for determining what constitutes conformance to this requirement.

2158 **8.7.1.4 Action APE_ECD.1.2E**

2159 **8.7.1.4.1 Work unit APE_ECD.1-13**

2160 The evaluator *shall examine* the extended components definition to determine that each extended
2161 component may not be clearly expressed using existing components.

2162 If the PP does not contain extended security requirements, this work unit is not applicable and
2163 therefore considered to be satisfied.

2164 The evaluator should take components from ISO/IEC 15408-2 and ISO/IEC 15408-3, other
2165 extended components that have been defined in the PP, combinations of these components, and
2166 possible operations on these components into account when making this determination.

2167 The evaluator is reminded that the role of this work unit is to preclude unnecessary duplication of
2168 components, that is, components that may be clearly expressed by using other components. The
2169 evaluator should not undertake an exhaustive search of all possible combinations of components
2170 including operations in an attempt to find a way to express the extended component by using
2171 existing components.

2172 **8.8 Security requirements (APE_REQ)**

2173 **8.8.1 Evaluation of sub-activity (APE_REQ.1)**

2174 **8.8.1.1 Objectives**

2175 The objective of this sub-activity is to determine whether the SFRs and SARs are clear,
2176 unambiguous and well-defined, whether they are internally consistent, and whether the SFRs
2177 counter the threats and implement the organisational security policies of the TOE.

2178 **8.8.1.2 Input**

2179 The evaluation evidence for this sub-activity is:

2180 a) the PP.

2181 **8.8.1.3 Action APE_REQ.1.1E**

2182 ISO/IEC 15408-3 APE_REQ.1.1C: *The statement of security requirements shall describe the SFRs and*
2183 *the SARs.*

2184 **8.8.1.3.1 Work unit APE_REQ.1-1**

2185 The evaluator ***shall check*** that the statement of security requirements describes the SFRs.

2186 The evaluator determines that each SFR is identified by one of the following means:

2187 a) by reference to an individual component in ISO/IEC 15408-2;

2188 b) by reference to an extended component in the extended components definition of the PP;

2189 c) by reference to a PP that the PP claims to be conformant with;

2190 d) by reference to a security requirements package that the PP claims to be conformant
2191 with;

2192 e) by reproduction in the PP.

2193 It is not required to use the same means of identification for all SFRs.

2194 **8.8.1.3.2 Work unit APE_REQ.1-2**

2195 The evaluator ***shall check*** that the statement of security requirements describes the SARs.

2196 The evaluator determines that each SAR is identified by one of the following means:

2197 a) by reference to an individual component in ISO/IEC 15408-3;

2198 b) by reference to an extended component in the extended components definition of the PP;

2199 c) by reference to a PP that the PP claims to be conformant with;

2200 d) by reference to a security requirements package that the PP claims to be conformant
2201 with;

2202 e) by reproduction in the PP.

- 2203 It is not required to use the same means of identification for all SARs.
- 2204 ISO/IEC 15408-3 APE_REQ.1.2C: *All subjects, objects, operations, security attributes, external entities*
2205 *and other terms that are used in the SFRs and the SARs shall be defined.*
- 2206 **8.8.1.3.3 Work unit APE_REQ.1-3**
- 2207 The evaluator ***shall examine*** the PP to determine that all subjects, objects, operations, security
2208 attributes, external entities and other terms that are used in the SFRs and the SARs are defined.
- 2209 The evaluator determines that the PP defines all:
- 2210 — (types of) subjects and objects that are used in the SFRs;
- 2211 — (types of) security attributes of subjects, users, objects, information, sessions and/or resources,
2212 possible values that these attributes may take and any relations between these values (e.g.
2213 top_secret is “higher” than secret);
- 2214 — (types of) operations that are used in the SFRs, including the effects of these operations;
- 2215 — (types of) external entities in the SFRs;
- 2216 — other terms that are introduced in the SFRs and/or SARs by completing operations, if these
2217 terms are not immediately clear, or are used outside their dictionary definition.
- 2218 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no
2219 misunderstanding may occur due to the introduction of vague terms. This work unit should not be
2220 taken into extremes, by forcing the PP writer to define every single word. The general audience of a
2221 set of security requirements should be assumed to have a reasonable knowledge of IT, security and
2222 “Evaluation criteria for IT security”.
- 2223 All of the above may be presented in groups, classes, roles, types or other groupings or
2224 characterisations that allow easy understanding.
- 2225 The evaluator is reminded that these lists and definitions do not have to be part of the statement of
2226 security requirements, but may be placed (in part or in whole) in different subclauses. This may be
2227 especially applicable if the same terms are used in the rest of the PP.
- 2228 ISO/IEC 15408-3 APE_REQ.1.3C: *The statement of security requirements shall include a natural*
2229 *language description, part of which describes how the SFRs combine together to provide security*
2230 *functionality in terms of the architecture that is visible to Administrators and other users.*
- 2231 **8.8.1.3.4 Work unit APE_REQ.1-4**
- 2232 The evaluator ***shall check*** that the statement of security requirements includes a natural language
2233 description, part of which describes how the SFRs combine together to provide security
2234 functionality in terms of the architecture that is visible to Administrators and other users.
- 2235 The description is intended to make clear connections between SFRs and to provide a view of how
2236 they provide security functionality that is recognizable to Administrators and other types of user.
2237 The description in terms of the architecture that is “visible to Administrators and other users”
2238 means that the description must relate the security behavior to visible elements, but the
2239 mechanisms themselves need not be visible. For example: when describing authentication using a
2240 biometric mechanism, the calculation of the match or score might not be visible, but (a) might
2241 relate to a referenced description of a matching algorithm, (b) might be based on specific template
2242 files maintained by the Administrator, and (c) will result in acceptance or rejection of the
2243 authentication attempt – therefore the description might make use of any or all of these items (a) –

- 2244 (c). No specific format for this information is prescribed, and the description need not all be located
 2245 alongside the SFRs themselves (e.g. some of it might be in the PP Introduction). The intention of the
 2246 requirement is to make the meaning of the SFRs clearer and more easily understood by readers of
 2247 the PP who may not have deep knowledge of the CC but who are familiar with the product type.
- 2248 The evaluator determines that all operations are identified in each SFR or SAR where such an
 2249 operation is used. This includes both completed operations and uncompleted operations.
 2250 Identification may be achieved by typographical distinctions, or by explicit identification in the
 2251 surrounding text, or by any other distinctive means.
- 2252 ISO/IEC 15408-3 APE_REQ.1.4C: *The statement of security requirements shall identify all operations*
 2253 *on the security requirements.*
- 2254 **8.8.1.3.5 Work unit APE_REQ.1-5**
- 2255 The evaluator ***shall check*** that the statement of security requirements identifies all operations on
 2256 the security requirements.
- 2257 The evaluator determines that all operations are identified in each SFR or SAR where such an
 2258 operation is used. This includes both completed operations and uncompleted operations.
 2259 Identification may be achieved by typographical distinctions, or by explicit identification in the
 2260 surrounding text, or by any other distinctive means.
- 2261 *If the PP defines selection-based SFRs, the evaluator determines that the PP clearly identifies the*
 2262 *dependencies between the selection in an SFR and the selection-based SFR(s) to be included in the*
 2263 *PP/ST should that selection be chosen by the PP/ST author.*
- 2264 ISO/IEC 15408-3 APE_REQ.1.5C: *All operations shall be performed correctly.*
- 2265 **8.8.1.3.6 Work unit APE_REQ.1-6**
- 2266 The evaluator ***shall examine*** the statement of security requirements to determine that all
 2267 assignment operations are performed correctly.
- 2268 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
 2269 **Guidance for Operations.**
- 2270 **8.8.1.3.7 Work unit APE_REQ.1-7**
- 2271 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration
 2272 operations are performed correctly.
- 2273 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
 2274 **Guidance for Operations.**
- 2275 **8.8.1.3.8 Work unit APE_REQ.1-8**
- 2276 The evaluator ***shall examine*** the statement of security requirements to determine that all selection
 2277 operations are performed correctly.
- 2278 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
 2279 **Guidance for Operations.**
- 2280 **8.8.1.3.9 Work unit APE_REQ.1-9**
- 2281 The evaluator ***shall examine*** the statement of security requirements to determine that all
 2282 refinement operations are performed correctly.

2283 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
2284 **Guidance for Operations**.

2285 ISO/IEC 15408-3 APE_REQ.1.6C: *Each dependency of the security requirements shall either be*
2286 *satisfied, or the security requirements rationale shall justify the dependency not being satisfied.*

2287 **8.8.1.3.10 Work unit APE_REQ.1-10**

2288 The evaluator **shall examine** the statement of security requirements to determine that each
2289 dependency of the security requirements is either satisfied, or that the security requirements
2290 rationale justifies the dependency not being satisfied.

2291 A dependency is satisfied by the inclusion of the relevant component (or one that is hierarchical to
2292 it) within the statement of security requirements. The component used to satisfy the dependency
2293 should, if necessary, be modified by operations to ensure that it actually satisfies that dependency.

2294 A justification that a dependency is not met should address either:

2295 a) why the dependency is not necessary or useful, in which case no further information is
2296 required; or

2297 b) that the dependency has been addressed by the operational environment of the TOE, in
2298 which case the justification should describe how the security objectives for the
2299 operational environment address this dependency.

2300 ISO/IEC 15408-3 APE_REQ.1.7C: *The security requirements rationale shall trace each SFR back to*
2301 *the threats countered by that SFR and OSPs enforced by that SFR.*

2302 **8.8.1.3.11 Work unit APE_REQ.1-11**

2303 The evaluator **shall check** that the security requirements rationale traces each SFR back to the
2304 threats countered by that SFR and OSPs enforced by that SFR.

2305 The evaluator determines that each SFR is traced back to at least one threat or OSP for the TOE.

2306 Failure to trace implies that either the security requirements rationale is incomplete, the security
2307 objectives for the TOE are incomplete, or the SFR has no useful purpose.

2308 There is no prescribed location for this part of the rationale: for example, the relevant parts may be
2309 located under each threat and OSP in order to help make the security argument clearer and easier
2310 to read.

2311 ISO/IEC 15408-3 APE_REQ.1.8C: *The security requirements rationale shall trace each security*
2312 *objective for the operational environment back to threats countered by that security objective, OSPs*
2313 *enforced by that security objective, and assumptions upheld by that security objective.*

2314 **8.8.1.3.12 Work unit APE_REQ.1-12**

2315 The evaluator **shall check** that the security objectives requirements rationale traces the security
2316 objectives for the operational environment back to threats countered by that security objective, to
2317 OSPs enforced by that security objective, and to assumptions upheld by that security objective.

2318 Each security objective for the operational environment may trace back to threats, OSPs,
2319 assumptions, or a combination of threats, OSPs and/or assumptions, but it must trace back to at
2320 least one threat, OSP or assumption.

2321 Failure to trace implies that either the security objectives requirements rationale is incomplete, the
 2322 security problem definition is incomplete, or the security objective for the operational
 2323 environment has no useful purpose.

2324 There is no prescribed location for this part of the rationale: for example, the relevant parts may be
 2325 located under each threat, OSP and assumption in order to help make the security argument
 2326 clearer and easier to read.

2327 ISO/IEC 15408-3 APE_REQ.1.9C: *The security requirements rationale shall demonstrate that the*
 2328 *SFRs (in conjunction with the security objectives for the environment) counter all threats for the TOE.*

2329 **8.8.1.3.13 Work unit APE_REQ.1-13**

2330 The evaluator **shall examine** the security requirements rationale to determine that for each threat
 2331 it demonstrates that the SFRs are suitable to meet that threat.

2332 If no SFRs trace back to a threat, the evaluator action related to this work unit is assigned a fail
 2333 verdict.

2334 The evaluator determines that the justification for a threat shows whether the threat is removed,
 2335 diminished or mitigated.

2336 The evaluator determines that the justification for a threat demonstrates that the SFRs are
 2337 sufficient: if all SFRs that trace back to the threat are achieved then, in the context of any applicable
 2338 OSPs and assumptions, the threat is removed, sufficiently diminished, or the effects of the threat
 2339 are sufficiently mitigated.

2340 Note that simply listing in the security requirements rationale the SFRs associated with each threat
 2341 may be part of a justification, but does not constitute a justification by itself. A descriptive
 2342 justification is required, although in simple cases this justification may be as minimal as "SFR X
 2343 directly counters Threat Y".

2344 The evaluator also determines that each SFR that traces back to a threat is necessary: when the SFR
 2345 is implemented it actually contributes to the removal, diminishing or mitigation of that threat.

2346 ISO/IEC 15408-3 APE_REQ.1.10C: *The security requirements rationale shall demonstrate that the*
 2347 *SFRs (in conjunction with the security objectives for the environment) enforce all OSPs for the TOE.*

2348 **8.8.1.3.14 Work unit APE_REQ.1-14**

2349 The evaluator **shall examine** the security requirements rationale to determine that for each OSP it
 2350 justifies that the SFRs are suitable to enforce that OSP.

2351 If no SFRs or security objectives for the operational environment trace back to the OSP, the
 2352 evaluator action related to this work unit is assigned a fail verdict.

2353 The evaluator determines that the justification for an OSP demonstrates that the security
 2354 objectives are sufficient: if all SFRs that trace back to that OSP are achieved then, in the context of
 2355 any applicable assumptions, the OSP is enforced.

2356 The evaluator also determines that each SFR that traces back to an OSP is necessary: when the SFR
 2357 is implemented it actually contributes to the enforcement of the OSP.

2358 Note that simply listing in the security requirements rationale the SFRs associated with each OSP
 2359 may be part of a justification, but does not constitute a justification by itself. A descriptive
 2360 justification is required, although in simple cases this justification may be as minimal as "SFR X
 2361 directly enforces OSP Y".

2362 ISO/IEC 15408-3 APE_REQ.1.11C: The security requirements rationale shall demonstrate that the
2363 SFRs (in conjunction with the security objectives for the environment) enforce all OSPs for the TOE.

2364 **8.8.1.3.15 Work unit APE_REQ.1-15**

2365 The evaluator *shall examine* the security requirements rationale to determine that for each
2366 assumption for the operational environment it contains an appropriate justification that the
2367 security objectives for the operational environment are suitable to uphold that assumption.

2368 If no security objectives for the operational environment trace back to the assumption, the
2369 evaluator action related to this work unit is assigned a fail verdict.

2370 The evaluator determines that the justification for an assumption about the operational
2371 environment of the TOE demonstrates that the security objectives are sufficient: if all security
2372 objectives for the operational environment that trace back to that assumption are achieved, the
2373 operational environment upholds the assumption.

2374 The evaluator also determines that each security objective for the operational environment that
2375 traces back to an assumption about the operational environment of the TOE is necessary: when the
2376 security objective is achieved it actually contributes to the operational environment upholding the
2377 assumption.

2378 Note that simply listing in the security requirements rationale the security objectives for the
2379 operational environment associated with each assumption may be a part of a justification, but does
2380 not constitute a justification by itself. A descriptive justification is required, although in simple
2381 cases this justification may be as minimal as "Security Objective X directly upholds Assumption Y".

2382

2383 ISO/IEC 15408-3 APE_REQ.1.12C: *The statement of security requirements shall be internally*
2384 *consistent.*

2385 **8.8.1.3.16 Work unit APE_REQ.1-16**

2386 The evaluator *shall examine* the statement of security requirements to determine that it is
2387 internally consistent.

2388 The evaluator determines that the combined set of all SFRs and SARs is internally consistent.

2389 The evaluator determines that on all occasions where different security requirements apply to the
2390 same types of developer evidence, events, operations, data, tests to be performed etc. or to "all
2391 objects", "all subjects" etc., that these requirements do not conflict.

2392 Some possible conflicts are:

2393 a) an extended SAR specifying that the design of a certain cryptographic algorithm is to be
2394 kept secret, and another extended SAR specifying an open source review;

2395 b) **FAU_GEN.1 Audit data generation** specifying that subject identity is to be logged,
2396 **FDP_ACC.1 Subset access control** specifying who has access to these logs, and **FPR_UNO.1**
2397 **Unobservability** specifying that some actions of subjects should be unobservable to other
2398 subjects. If the subject that should not be able to see an activity may access logs of this
2399 activity, these SFRs conflict;

2400 c) **FDP_RIP.1 Subset residual information protection** specifying deletion of information no
2401 longer needed, and **FDP_ROL.1 Basic rollback** specifying that a TOE may return to a
2402 previous state. If the information that is needed for the rollback to the previous state has
2403 been deleted, these requirements conflict;

- 2404 d) Multiple iterations of **FDP_ACC.1 Subset access control** especially where some iterations
 2405 cover the same subjects, objects, or operations. If one access control SFR allows a subject
 2406 to perform an operation on an object, while another access control SFR does not allow
 2407 this, these requirements conflict.

2408 **8.8.2 Evaluation of sub-activity (APE_REQ.2)**

2409 **8.8.2.1 Objectives**

2410 The objective of this sub-activity is to determine whether the SFRs and SARs are clear,
 2411 unambiguous and well-defined, whether they are internally consistent, and whether the SFRs meet
 2412 the security objectives of the TOE.

2413 **8.8.2.2 Input**

2414 The evaluation evidence for this sub-activity is:

- 2415 a) the PP.

2416 **8.8.2.3 Action APE_REQ.2.1E**

2417 ISO/IEC 15408-3 APE_REQ.2.1C: *The statement of security requirements shall describe the SFRs and*
 2418 *the SARs.*

2419 **8.8.2.3.1 Work unit APE_REQ.2-1**

2420 The evaluator **shall check** that the statement of security requirements describes the SFRs.

2421 The evaluator determines that each SFR is identified by one of the following means:

- 2422 a) by reference to an individual component in ISO/IEC 15408-2;
 2423 b) by reference to an extended component in the extended components definition of the PP;
 2424 c) by reference to an individual component in a PP that the PP claims to be conformant with;
 2425 d) by reference to an individual component in a security requirements package that the PP
 2426 claims to be conformant with;
 2427 e) by reproduction in the PP.

2428 It is not required to use the same means of identification for all SFRs.

2429 **8.8.2.3.2 Work unit APE_REQ.2-2**

2430 The evaluator **shall check** that the statement of security requirements describes the SARs.

2431 The evaluator determines that each SAR is identified by one of the following means:

- 2432 a) by reference to an individual component in ISO/IEC 15408-3;
 2433 b) by reference to an extended component in the extended components definition of the PP;
 2434 c) by reference to an individual component in a PP that the PP claims to be conformant with;
 2435 d) by reference to an individual component in a security requirements package that the PP
 2436 claims to be conformant with;

2437 e) by reproduction in the PP.

2438 It is not required to use the same means of identification for all SARs.

2439 ISO/IEC 15408-3 APE_REQ.2.2C: *All subjects, objects, operations, security attributes, external entities*
2440 *and other terms that are used in the SFRs and the SARs shall be defined.*

2441 **8.8.2.3.3 Work unit APE_REQ.2-3**

2442 The evaluator **shall examine** the PP to determine that all subjects, objects, operations, security
2443 attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

2444 The evaluator determines that the PP defines all:

2445 — (types of) subjects and objects that are used in the SFRs;

2446 — (types of) security attributes of subjects, users, objects, information, sessions and/or resources,
2447 possible values that these attributes may take and any relations between these values (e.g.
2448 top_secret is “higher” than secret);

2449 — (types of) operations that are used in the SFRs, including the effects of these operations;

2450 — (types of) external entities in the SFRs;

2451 — other terms that are introduced in the SFRs and/or SARs by completing operations, if these
2452 terms are not immediately clear, or are used outside their dictionary definition.

2453 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no
2454 misunderstanding may occur due to the introduction of vague terms. This work unit should not be
2455 taken into extremes, by forcing the PP writer to define every single word. The general audience of a
2456 set of security requirements should be assumed to have a reasonable knowledge of IT, security and
2457 “Evaluation criteria for IT security”.

2458 All of the above may be presented in groups, classes, roles, types or other groupings or
2459 characterisations that allow easy understanding.

2460 The evaluator is reminded that these lists and definitions do not have to be part of the statement of
2461 security requirements, but may be placed (in part or in whole) in different subclauses. This may be
2462 especially applicable if the same terms are used in the rest of the PP.

2463 ISO/IEC 15408-3 APE_REQ.2.3C: *The statement of security requirements shall identify all operations*
2464 *on the security requirements.*

2465 **8.8.2.3.4 Work unit APE_REQ.2-4**

2466 The evaluator **shall check** that the statement of security requirements identifies all operations on
2467 the security requirements.

2468 The evaluator determines that all operations are identified in each SFR or SAR where such an
2469 operation is used. This includes both completed operations and uncompleted operations.
2470 Identification may be achieved by typographical distinctions, or by explicit identification in the
2471 surrounding text, or by any other distinctive means.

2472 *If the PP defines selection-based SFRs, the evaluator determines that the PP clearly identifies the*
2473 *dependencies between the selection in an SFR and the selection-based SFR(s) to be included in the*
2474 *PP/ST should that selection be chosen by the PP/ST author.*

- 2475 ISO/IEC 15408-3 APE_REQ.2.4C: *All operations shall be performed correctly.*
- 2476 **8.8.2.3.5 Work unit APE_REQ.2-5**
- 2477 The evaluator ***shall examine*** the statement of security requirements to determine that all
2478 assignment operations are performed correctly.
- 2479 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
2480 **Guidance for Operations.**
- 2481 **8.8.2.3.6 Work unit APE_REQ.2-6**
- 2482 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration
2483 operations are performed correctly.
- 2484 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
2485 **Guidance for Operations.**
- 2486 **8.8.2.3.7 Work unit APE_REQ.2-7**
- 2487 The evaluator ***shall examine*** the statement of security requirements to determine that all selection
2488 operations are performed correctly.
- 2489 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
2490 **Guidance for Operations.**
- 2491 **8.8.2.3.8 Work unit APE_REQ.2-8**
- 2492 The evaluator ***shall examine*** the statement of security requirements to determine that all
2493 refinement operations are performed correctly.
- 2494 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
2495 **Guidance for Operations.**
- 2496 ISO/IEC 15408-3 APE_REQ.2.5C: *Each dependency of the security requirements shall either be*
2497 *satisfied, or the security requirements rationale shall justify the dependency not being satisfied.*
- 2498 **8.8.2.3.9 Work unit APE_REQ.2-9**
- 2499 The evaluator ***shall examine*** the statement of security requirements to determine that each
2500 dependency of the security requirements is either satisfied, or that the security requirements
2501 rationale justifies the dependency not being satisfied.
- 2502 A dependency is satisfied by the inclusion of the relevant component (or one that is hierarchical to
2503 it) within the statement of security requirements. The component used to satisfy the dependency
2504 should, if necessary, be modified by operations to ensure that it actually satisfies that dependency.
- 2505 A justification that a dependency is not met should address either:
- 2506 a) why the dependency is not necessary or useful, in which case no further information is
2507 required; or
- 2508 b) that the dependency has been addressed by the operational environment of the TOE, in
2509 which case the justification should describe how the security objectives for the
2510 operational environment address this dependency.
- 2511 ISO/IEC 15408-3 APE_REQ.2.6C: *The security requirements rationale shall trace each SFR back to*
2512 *the security objectives for the TOE.*

2513 **8.8.2.3.10 Work unit APE_REQ.2-10**

2514 The evaluator **shall check** that the security requirements rationale traces each SFR back to the
2515 security objectives for the TOE.

2516 The evaluator determines that each SFR is traced back to at least one security objective for the TOE.

2517 Failure to trace implies that either the security requirements rationale is incomplete, the security
2518 objectives for the TOE are incomplete, or the SFR has no useful purpose.

2519 ISO/IEC 15408-3 APE_REQ.2.7C: *The security requirements rationale shall demonstrate that the*
2520 *SFRs meet all security objectives for the TOE.*

2521 **8.8.2.3.11 Work unit APE_REQ.2-11**

2522 The evaluator **shall examine** the security requirements rationale to determine that for each
2523 security objective for the TOE it justifies that the SFRs are suitable to meet that security objective
2524 for the TOE.

2525 If no SFRs trace back to the security objective for the TOE, the evaluator action related to this work
2526 unit is assigned a fail verdict.

2527 The evaluator determines that the justification for a security objective for the TOE demonstrates
2528 that the SFRs are sufficient: if all SFRs that trace back to the objective are satisfied, the security
2529 objective for the TOE is achieved.

2530 If the SFRs that trace back to a security objective for the TOE have any uncompleted assignments,
2531 or uncompleted or restricted selections, the evaluator determines that for every conceivable
2532 completion or combination of completions of these operations, the security objective is still met.

2533 The evaluator also determines that each SFR that traces back to a security objective for the TOE is
2534 necessary: when the SFR is satisfied, it actually contributes to achieving the security objective.

2535 Note that the tracings from SFRs to security objectives for the TOE provided in the security
2536 requirements rationale may be a part of the justification, but do not constitute a justification by
2537 themselves.

2538 ISO/IEC 15408-3 APE_REQ.2.8C: *The security requirements rationale shall explain why the SARs*
2539 *were chosen.*

2540 **8.8.2.3.12 Work unit APE_REQ.2-12**

2541 The evaluator **shall check** that the security requirements rationale explains why the SARs were
2542 chosen.

2543 The evaluator is reminded that any explanation is correct, as long as it is coherent and neither the
2544 SARs nor the explanation have obvious inconsistencies with the remainder of the PP.

2545 An example of an obvious inconsistency between the SARs and the remainder of the PP would be to
2546 have threat agents that are very capable, but an AVA_VAN SAR that does not protect against these
2547 threat agents.

2548 ISO/IEC 15408-3 APE_REQ.2.9C: *The statement of security requirements shall be internally*
2549 *consistent.*

2550 **8.8.2.3.13 Work unit APE_REQ.2-13**

2551 The evaluator **shall examine** the statement of security requirements to determine that it is
2552 internally consistent.

2553 The evaluator determines that the combined set of all SFRs and SARs is internally consistent.

2554 The evaluator determines that on all occasions where different security requirements apply to the
2555 same types of developer evidence, events, operations, data, tests to be performed etc. or to “all
2556 objects”, “all subjects” etc., that these requirements do not conflict.

2557 Some possible conflicts are:

2558 c) an extended SAR specifying that the design of a certain cryptographic algorithm is to be
2559 kept secret, and another extended SAR specifying an open source review;

2560 d) **FAU_GEN.1 Audit data generation** specifying that subject identity is to be logged,
2561 **FDP_ACC.1 Subset access control** specifying who has access to these logs, and **FPR_UNO.1**
2562 **Unobservability** specifying that some actions of subjects should be unobservable to other
2563 subjects. If the subject that should not be able to see an activity may access logs of this
2564 activity, these SFRs conflict;

2565 e) **FDP_RIP.1 Subset residual information protection** specifying deletion of information no
2566 longer needed, and **FDP_ROL.1 Basic rollback** specifying that a TOE may return to a
2567 previous state. If the information that is needed for the rollback to the previous state has
2568 been deleted, these requirements conflict;

2569 Multiple iterations of **FDP_ACC.1 Subset access control** especially where some iterations
2570 cover the same subjects, objects, or operations. If one access control SFR allows a subject
2571 to perform an operation on an object, while another access control SFR does not allow
2572 this, these requirements conflict.

2573 **9 Class ACE: Protection Profile Configuration evaluation**

2574 **9.1 Introduction**

2575 All Base-PP(s) referenced in the PP-Module must be evaluated before the evaluation of a PP-
2576 Configuration.

2577 One possibility for evaluating a PP-Configuration is to flatten/serialise all the components of the
2578 Base-PP(s) and PP-Modules composing the PP-Configuration, duplicating components as necessary,
2579 and evaluating the resulting PP as a standard PP.

2580 Another possibility for evaluation of a PP-Configuration composed of several PP-Modules proceeds
2581 PP-Module by PP-Module, iteratively. Considering a PP-Configuration composed of the Protection
2582 Profiles P_i and the PP-Modules M_j , evaluation of the PP-Configuration proceeds with the following
2583 steps, illustrated in Figure 6

2584 1) first evaluating independently all Protection Profiles P_i ;

2585 2) evaluating the PP-Configuration C_1 composed of the PP-Module M_1 with the Protection
2586 Profiles P_i ;

2587 3) evaluating the PP-Configuration C_{i+1} composed of the PP-Module M_{i+1} with the PP-
2588 Configuration C_i considered as a standard PP (cf. Section B.14 in ISO/IEC 15408-1);

2589 4) iterating the step 3 for all the PP-Modules

2590 Steps 2 and 3 are themselves performed in two steps:

- 2591 a) Evaluation of the PP-Module with its Base-PP(s) (Evaluation of sub-activity (ACE_MCO.1))
- 2592 b) Extension of the evaluation (consistency assessment) to the other elements of the PP-
- 2593 Configuration (Evaluation of sub-activity (ACE_CCO.1))

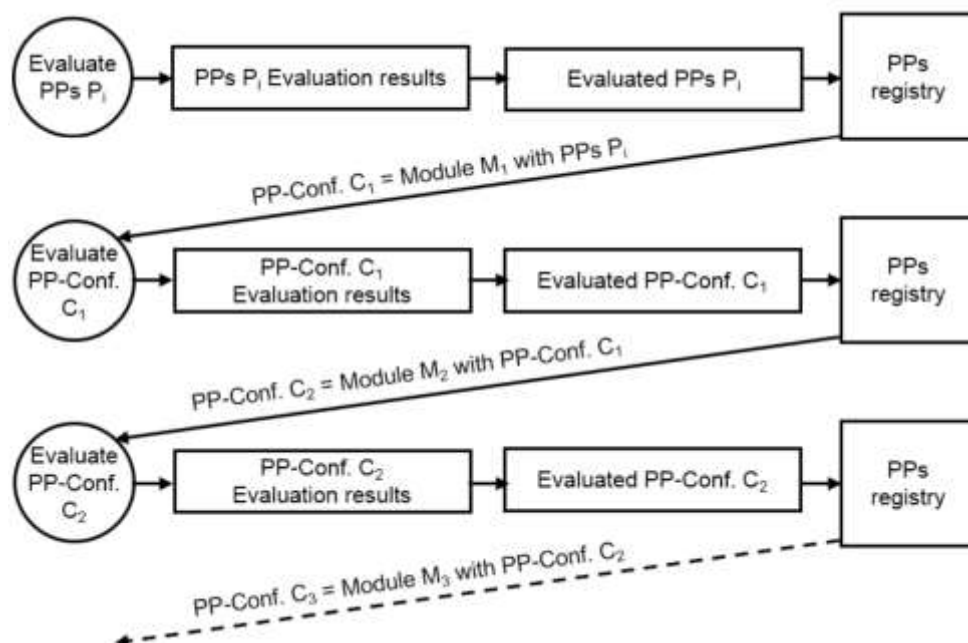


Figure 6 - Evaluation of a PP-Configuration

2597 The ACE evaluation methodology is based on APE's. The common parts are not duplicated in this

2598 document but referred to.

2599 9.2 PP-Module introduction (ACE_INT)

2600 9.2.1 Evaluation of sub-activity (ACE_INT.1)

2601 9.2.1.1 Objectives

2602 The objective of this sub-activity is to determine whether the PP-Module is correctly identified, and

2603 whether the Base-PP(s) and TOE overview are consistent with each other.

2604 9.2.1.2 Input

2605 The evaluation evidence for this sub-activity is:

- 2606 a) the PP-Module;
- 2607 b) its Base-PP(s)

2608 9.2.1.3 Application notes

2609 All actions of APE_INT.1.1E hold.

2610 **9.2.1.4 Action ACE_INT.1.1E**

2611 ISO/IEC 15408-3 ACE_INT.1.1C *The PP-Module introduction shall uniquely identify all the Base-PPs*
 2612 *on which the PP-Module relies, including their logical structuring and relationship to the PP-Module*
 2613 *according to ISO/IEC 15408 Part 1, section B.13.3.2.*

2614 **9.2.1.4.1 Work unit ACE_INT.1-1**

2615 *The evaluator **shall check** that the PP-Module introduction identifies the Base-PP(s) on which the PP-*
 2616 *Module relies.*

2617 ISO/IEC 15408-3 ACE_INT.1.2C *The TOE overview shall identify the differences introduced by the PP-*
 2618 *Module with respect to the TOE overview of its Base-PP(s).*

2619 **9.2.1.4.2 Work unit ACE_INT.1-2**

2620 The evaluator **shall check** that the TOE overview identifies the differences introduced by the PP-
 2621 Module with respect to the TOE overview of its Base-PP(s).

2622 **9.3 PP-Module conformance claims (ACE_CCL)**

2623 **9.3.1 Evaluation of sub-activity (ACE_CCL.1)**

2624 **9.3.1.1 Objectives**

2625 The objective of this sub-activity is to determine the validity of various conformance claims. These
 2626 describe how the PP-Module conforms to the ISO/IEC 15408 Part 2 and SFR packages.

2627 **9.3.1.2 Input**

2628 The evaluation evidence for this sub-activity is:

- 2629 a) the PP-Module;
- 2630 b) the SFR package(s) that the PP claims conformance to;
- 2631 c) the PP-Configuration.

2632 **9.3.1.3 Action ACE_CCL.1.1E**

2633 ISO/IEC 15408-3 ACE_CCL.1.1C *The conformance claim shall contain a ISO/IEC 15408*
 2634 *conformance claim that identifies the version of the ISO/IEC 15408 to which the PP-Module*
 2635 *claims conformance.*

2636 **9.3.1.3.1 Work unit ACE_CCL.1-1**

2637 The evaluator **shall check** that the conformance claim contains a ISO/IEC 15408 conformance
 2638 claim that identifies the version of the ISO/IEC 15408 to which the PP-Module claims
 2639 conformance.

2640 The evaluator determines that the ISO/IEC 15408 conformance claim identifies the version of
 2641 the ISO/IEC 15408 that was used to develop this PP-Module. This should include the version
 2642 number of the ISO/IEC 15408 and, unless the International English version of the ISO/IEC
 2643 15408 was used, the language of the version of the ISO/IEC 15408 that was used.

2644 ISO/IEC 15408-3 ACE_CCL.1.2C *The ISO/IEC 15408 conformance claim shall describe the*
 2645 *conformance of the PP-Module to ISO/IEC 15408 Part 2 as either ISO/IEC 15408 Part 2*
 2646 *conformant or ISO/IEC 15408 Part 2 extended.*

2647 **9.3.1.3.2 Work unit ACE_CCL.1-2**

2648 The evaluator **shall check** that the ISO/IEC 15408 conformance claim states a claim of either
2649 ISO/IEC 15408 Part 2 conformant or ISO/IEC 15408 Part 2 extended for the PP-Module.

2650 ISO/IEC 15408-3 ACE_CCL.1.3C *The conformance claim shall identify all security functional*
2651 *requirement packages to which the PP-Module claims conformance.*

2652 **9.3.1.3.3 Work unit ACE_CCL.1-3**

2653 The evaluator **shall check** that, for each identified package, the conformance claim contains a
2654 package claim that identifies all security functional requirement packages to which the PP-
2655 Module claims conformance.

2656 If the PP-Module does not claim conformance to a security functional requirement package,
2657 this work unit is not applicable and therefore considered to be satisfied.

2658 The evaluator determines that any referenced security functional requirement packages are
2659 unambiguously identified (e.g. by title and version number, or by the identification included in
2660 the introduction of that security functional requirement package).

2661 The evaluator is reminded that claims of partial conformance to a security functional
2662 requirement package are not permitted.

2663 ISO/IEC 15408-3 ACE_CCL.1.4C *The ISO/IEC 15408 conformance claim shall be consistent with*
2664 *the extended components definition.*

2665 **9.3.1.3.4 Work unit ACE_CCL.1-4**

2666 The evaluator **shall examine** the ISO/IEC 15408 conformance claim for ISO/IEC 15408 Part 2
2667 to determine that it is consistent with the extended components definition

2668 If the ISO/IEC 15408 conformance claim contains ISO/IEC 15408 Part 2 conformant, the
2669 evaluator determines that the extended components definition does not define functional
2670 components.

2671 If the ISO/IEC 15408 conformance claim contains ISO/IEC 15408 Part 2 extended, the
2672 evaluator determines that the extended components definition defines at least one extended
2673 functional component.

2674 ISO/IEC 15408-3 ACE_CCL.1.5C *The conformance statement shall identify other PP-modules (if*
2675 *any) and PPs (that are not Base-PPs for the PP-Module under evaluation) that, in combination*
2676 *with the module under evaluation, can be used in a PP-configuration.*

2677 **9.3.1.3.5 Work unit ACE_CCL.1-5**

2678 The evaluator **shall check** the conformance statement to determine that it lists the set of other
2679 PP-modules that can be specified in the components statement of a PP-configuration that
2680 includes the PP-module.

2681 If no PPs in the PP-Configuration's component statement require exact conformance in their
2682 conformance statements then this work unit does not apply and is therefore considered
2683 satisfied.

2684 If the PP-module does not allow its use (in a PP-configuration) with other PP-modules, then
2685 there will be no other PP-modules identified in the PP-module's conformance statement, and
2686 the evaluator ensures the PP-configuration contains no other PP-modules in the PP-
2687 configuration's components statement.

2688 If the PP-configuration's components statement does include other PP-modules, then the
 2689 evaluator ensures that all PP-modules listed in the PP-configuration's components statement
 2690 are identified as allowed with the PP-module in its conformance statement.

2691 **9.3.1.3.6 Work unit ACE_CCL.1-6**

2692 The evaluator shall check the conformance statement to determine that it lists PPs identified
 2693 in the PP-Configuration's component statements that are not included in the PP-Module's set
 2694 of Base-PPs as identified in the PP-Configuration's component statements.

2695 If a PP in the PP-Configuration's component statement does not require exact conformance in
 2696 its conformance statement, this work unit does not apply and is therefore considered satisfied.

2697 If PP-Module does not identify (in its conformance statement) any PPs other than those that
 2698 make up the set of Base-PPs for the PP-Module identified in the PP-Configuration's component
 2699 statement, the evaluator ensures the PP-configuration contains no other (non-Base-) PPs in
 2700 the PP-configuration's components statement.

2701 If the PP-configuration's components statement does include PPs that are not part of the PP-
 2702 Module's set of Base-PPs, then the evaluator ensures that all such PPs listed in the PP-
 2703 configuration's components statement are identified as allowed with the PP-Module in its
 2704 conformance statement.

2705 ISO/IEC 15408-3 ACE_CCL.1.6C *The conformance claim shall describe any conformance of the*
 2706 *PP to a package as either package-conformant or package-augmented.*

2707 **9.3.1.3.7 Work unit ACE_CCL.1-7**

2708 The evaluator **shall check** that, for each identified package, the conformance claim states a claim of
 2709 either package-name conformant or package-name augmented.

2710 If the PP-Module does not claim conformance to a package, this work unit is not applicable and
 2711 therefore considered to be satisfied. PP-Modules can only claim conformance to functional
 2712 packages and therefore only this type of package is considered in the description below.

2713 If the functional package conformance claim contains package-name conformant, the evaluator
 2714 determines that all assumptions, threats, OSPs, security objectives and SFRs included in the
 2715 package are included in identical form by the PP-Module (including via its base-PP(s)).

2716 If the functional package conformance claim contains package-name augmented, the evaluator
 2717 determines that all all assumptions, threats, OSPs, security objectives and SFRs included in the
 2718 package are included in identical form by the PP-Module except that the PP-Module shall have at
 2719 least one additional SFR or one SFR that is hierarchically higher than an SFR in the functional
 2720 package.

2721 **9.4 PP-Module Security problem definition (ACE_SPD)**

2722 **9.4.1 Evaluation of sub-activity (ACE_SPD.1)**

2723 **9.4.1.1 Application notes**

2724 All actions of APE_SPD.1.1E hold.

2725 **9.5 PP-Module Security objectives (ACE_OBJ)**

2726 **9.5.1 Evaluation of sub-activity (ACE_OBJ.1)**

2727 **9.5.1.1 Application notes**

2728 If the PP-Configuration uses the Direct Rationale approach (as determined in ACE_CCO.1-2)
2729 then all actions of APE_OBJ.1.1E hold, otherwise all actions of APE_OBJ.2.1E hold.

2730 **9.6 PP-Module extended components definition (ACE_ECD)**

2731 **9.6.1 Evaluation of sub-activity (ACE_ECD.1)**

2732 **9.6.1.1 Application notes**

2733 All actions of APE_ECD.1.1E hold.

2734 **9.7 PP-Module security requirements (ACE_REQ)**

2735 **9.7.1 Evaluation of sub-activity (ACE_REQ.1)**

2736 **9.7.1.1 Application notes**

2737 If the PP-Configuration uses the Direct Rationale approach (as determined in ACE_CCO.1-2)
2738 then all actions of APE_REQ.1.1E hold, otherwise all actions of APE_REQ.2.1E hold. In either
2739 case the SAR part is not considered because it is empty in PP-Modules.

2740 **9.8 PP-Module consistency (ACE_MCO)**

2741 **9.8.1 Evaluation of sub-activity (ACE_MCO.1)**

2742 **9.8.1.1 Objectives**

2743 The objective of this sub-activity is to determine the consistency of the PP-Module regarding
2744 its Base-PP(s).

2745 **9.8.1.2 Input**

2746 The evaluation evidence for this sub-activity is:

2747 a) the PP-Module;

2748 b) its Base-PP(s)

2749 **9.8.1.3 Action ACE_MCO.1.1E**

2750 ISO/IEC 15408-3 ACE_MCO.1.1C The consistency rationale shall demonstrate that the TOE
2751 type of the PP-Module is consistent with the TOE type(s) in the Base-PPs identified in the PP-
2752 Module introduction.

2753 **9.8.1.3.1 Work unit ACE_MCO.1-1**

2754 The evaluator *shall examine* the consistency rationale to determine that the TOE type of the
2755 PP-Module is consistent with all the TOE types of the Base-PP(s).

2756 The relation between the types may be simple: a PP-Module may consider a TOE that provides
 2757 additional security functionality, or more complex: a TOE that provides a given security
 2758 functionality in a specific way.

2759 ISO/IEC 15408-3 ACE_MCO.1.2C The consistency rationale shall demonstrate that the
 2760 statement of the security problem definition is consistent with the statement of the security
 2761 problem definition in the Base-PPs identified in the PP-Module introduction.

2762 **9.8.1.3.2 Work unit ACE_MCO.1-2**

2763 The evaluator **shall examine** the PP-Module consistency rationale to determine that it
 2764 demonstrates that the statement of security problem definition of the PP-Module is consistent
 2765 with the statements of security problem definition stated in its Base-PPs.

2766 In particular, the evaluator examines the consistency rationale to determine that:

2767 a) the statements of threats, assumptions and OSPs in the PP-Module do not contradict those
 2768 from the Base-PP(s).

2769 b) the statement of assumptions in the PP-Module addresses aspects out of scope of the
 2770 Base-PP, in which case, the addition of elements is allowed.

2771 ISO/IEC 15408-3 ACE_MCO.1.3C The consistency rationale shall demonstrate that the
 2772 statement of security objectives is consistent with the statement of security objectives in the
 2773 Base-PPs identified in the PP-Module introduction.

2774 **9.8.1.3.3 Work unit ACE_MCO.1-3**

2775 The evaluator **shall examine** the PP-Module consistency rationale to determine that it
 2776 demonstrates that the statement of security objectives of the PP-Module is consistent with the
 2777 statement of security objectives of its Base-PP(s).

2778 Where the PP-Module and its Base-PP(s) use the Direct Rationale approach then this work
 2779 unit is trivially satisfied for the TOE objectives (because these are not included under the
 2780 Direct Rationale approach). If *any* of the PP-Module or its Base-PPs use the Direct Rationale
 2781 approach then the PP-Module *and all* of its Base-PPs must use the Direct Rationale approach,
 2782 otherwise the evaluator action related to this work unit is assigned a fail verdict.

2783 In particular, the evaluator examines the consistency rationale to determine that:

2784 a) the statements of the security objectives for the TOE and the security objectives for the
 2785 operational environment in the PP-Module do not contradict those from the Base-PPs.

2786 b) the statement of the security objectives for the operational environment in the PP-Module
 2787 addresses aspects out of scope of the Base-PP, in which case, the addition of elements is
 2788 allowed.

2789 ISO/IEC 15408-3 ACE_MCO.1.4C The consistency rationale shall demonstrate that the
 2790 statement of security requirements is consistent with the statement of security requirements
 2791 in the Base-PPs identified in the PP-Module introduction.

2792 **9.8.1.3.4 Work unit ACE_MCO.1-4**

2793 The evaluator **shall examine** the consistency rationale to determine that the statement of
 2794 security requirements of the PP-Module is consistent with the statement of security
 2795 requirements of its Base-PPs, that is, the SFRs of the PP-Module either complete or refine the
 2796 SFRs of the Base-PP(s) and that no contradiction arises from the whole set of SFRs of the PP-
 2797 Module and the Base-PP(s).

2798 **9.9 PP-Configuration consistency (ACE_CCO)**

2799 **9.9.1 Evaluation of sub-activity (ACE_CCO.1)**

2800 **9.9.1.1 Objectives**

2801 The objective of this sub-activity is to determine whether the PP-Configuration and its
2802 components are correctly identified.

2803 The objective of this sub-activity is also to determine the consistency of the PP-Configuration
2804 regarding the whole set of Protection Profiles and PP-Modules.

2805 For the consistency analysis required by this activity, the application notes of ISO/IEC 18045,
2806 Section 10.2.1 (Re-using the evaluation results of certified PPs), is applicable to determine
2807 which parts of the Base-PPs are to be re-evaluated during the evaluation of PP-Configuration.

2808 **9.9.1.2 Input**

2809 The evaluation evidence for this sub-activity is:

2810 a) the PP-Configuration reference;

2811 b) the PP-Configuration components statement;

2812 c) the PP(s) and PP-Modules identified in the components statement.

2813 **9.9.1.3 Action ACE_CCO.1.1E**

2814 ISO/IEC 15408-3 ACE_CCO.1.1C The PP-Configuration reference shall uniquely identify the
2815 PP-Configuration.

2816 **9.9.1.3.1 Work unit ACE_CCO.1-1**

2817 The evaluator shall examine the PP-Configuration reference to determine that it uniquely
2818 identifies the PP-Configuration.

2819 The evaluator determines that the PP-Configuration reference identifies the PP-Configuration
2820 itself, so that it may be easily distinguished from other PPs, PP-Configurations and PP-
2821 Modules, and that it also uniquely identifies each version of the PP-Configuration, e.g. by
2822 including a version number and/or a date of publication.

2823 The PP-Configuration should have some referencing system that is capable of supporting
2824 unique references (e.g. use of numbers, letters or dates).

2825 ISO/IEC 15408-3 ACE_CCO.1.2C The components statements shall uniquely identify the
2826 Protection Profiles and the PP-Modules that compose the PP-Configuration.

2827 **9.9.1.3.2 Work unit ACE_CCO.1-2**

2828 The evaluator shall examine the PP-Configuration components statement to determine that it
2829 uniquely identifies the Protection Profiles and PP-Modules contained in the PP-Configuration.

2830 The evaluator shall check that if *any* of the Base-PPs or PP-Modules in the PP-Configuration
2831 use the Direct Rationale Approach then *all* Base-PPs and PP-Modules in the PP-Configuration
2832 use the Direct Rationale approach.

2833 The Protection Profiles should have been certified and available for use in security targets.

2834 ISO/IEC 15408-3 ACE_CCO.1.3C The conformance statement shall specify the required
 2835 conformance to the PP-Configuration as one of exact, strict, or demonstrable. The
 2836 conformance claim shall contain a ISO/IEC 15408 conformance claim that identifies the
 2837 version of the ISO/IEC 15408 to which the PP-Configuration and its underlying Protection
 2838 Profiles and PP-Module claim conformance.

2839 **9.9.1.3.3 Work unit ACE_CCO.1-3**

2840 The evaluator shall examine the PP-Configuration conformance statement to determine that it
 2841 specifies the kind of conformance required: exact, strict, or demonstrable.

2842 The evaluator shall check that the conformance claim contains a ISO/IEC 15408 conformance
 2843 claim that identifies the version of the ISO/IEC 15408 to which the PP-Configuration and its
 2844 underlying Protection Profile(s) and PP-Module(s) claim conformance.

2845 The evaluator shall examine the PP-Configuration conformance claim to determine the
 2846 compatibility between all ISO/IEC 15408 versions that are related to the PP-Configuration
 2847 and its underlying Protection Profile(s) and PP-Module(s).

2848 **If at least one of the Protection Profiles identified in the PP-configuration components**
 2849 **statement requires exact conformance, then the PP-configuration conformance statement**
 2850 **shall also require exact conformance. If none of the PPs identified in the PP-configuration**
 2851 **components statement requires exact conformance but** at least one of the Protection Profiles
 2852 identified in the PP-Configuration components statement claims strict conformance, then the
 2853 PP-Configuration conformance statement shall also require strict conformance also.

2854 ISO/IEC 15408 versions used in a PP-Configuration and its underlying Protection Profile(s)
 2855 and PP-Module(s) have to be compatible. If compatibility is not obvious, guidance from the
 2856 certification scheme should be asked.

2857 ISO/IEC 15408-3 ACE_CCO.1.4C The SAR statement shall specify the set of SAR or predefined
 2858 EAL that applies to this PP-Configuration.

2859 **9.9.1.3.4 Work unit ACE_CCO.1-4**

2860 The evaluator shall examine the PP-Configuration SAR statement to determine that it specifies
 2861 a well-formed package of SAR. The SAR package can be built with components from ISO/IEC
 2862 15408-3 or can refer to a specific SAR package stated in one of the Protection Profiles
 2863 composing the PP-Configuration.

2864 If the set of SAR comes from ISO/IEC 15408-3 then the evaluator shall check that it is well-
 2865 formed: it is closed by dependencies or the SAR statements provide a sound discarding
 2866 rationale.

2867 The evaluator shall check that the set of SAR of the PP-Configuration is consistent with respect
 2868 to the SARs of each of the Protection Profiles contained in the PP-Configuration: for any SAR
 2869 component in each of the Protection Profile, the PP-Configuration provides either the same
 2870 component or a higher component in the family hierarchy. If the SAR component in the
 2871 Protection Profile is a refinement of a standard component, then the correspondent SAR
 2872 component in the PP-Configuration has to include these refinements. If two Protection
 2873 Profiles refine the same SAR component, the evaluator shall check that the refinements are
 2874 not contradictory and that the corresponding SAR component in the PP-Configuration meets
 2875 both.

2876 ISO/IEC 15408-3 ACE_CCO.1.5C The Base-PP(s) on which the PP-Modules relies shall belong
 2877 to the Protection Profiles identified in the components statement of the PP-Configuration.

2878 **9.9.1.3.5 Work unit ACE_CCO.1-5**

2879 The evaluator shall check that the Base-PP(s) of each PP-Module in the PP-Configuration are
2880 included in the set of Protection Profiles identified in the PP-Configuration's component
2881 statement. Where a PP-Module specifies alternative sets of Base-PP(s) then only one of these
2882 sets must be referred to in the PP-Configuration.

2883 ISO/IEC 15408-3 ACE_CCO.1.6C *The conformance statement of each Base-PPs and PP in the*
2884 *components statement of the PP-Configuration shall identify other PP-Modules and PPs that can*
2885 *be used in combination with the PP in a PP-Configuration.*

2886 **9.9.1.3.6 Work unit ACE_CCO.1-6**

2887 For each Protection Profile listed in the PP-Configuration's components statement, the
2888 evaluator shall check the PP's conformance statement to determine that all PP-modules
2889 specified in the PP-Configuration's components statement are listed as allowed to be used
2890 with that PP. If the PP-configuration does not require exact conformance in its conformance
2891 statement, this work unit does not apply and is therefore considered satisfied.

2892 The evaluator checks each PP in the PP-Configuration's components statement. For each PP,
2893 the evaluator determines that each PP-Module listed in the PP-Configuration's components
2894 statement is also listed in the PP's conformance statement as allowed to be used with that PP.

2895 **9.9.1.3.7 Work unit ACE_CCO.1-7**

2896 For each Protection Profile listed in the PP-Configuration's components statement, the
2897 evaluator shall check the PP's conformance statement to determine that all other PPs
2898 specified in the PP-Configuration's components statement are listed as allowed to be used
2899 with that PP.

2900 If the PP-Configuration does not require exact conformance in its conformance statement, this
2901 work unit does not apply and is therefore considered satisfied.

2902 If there is only one PP identified in the PP-Configuration's component statement, then this
2903 work unit does not apply and is therefore considered satisfied.

2904 **9.9.1.4 Action ACE_CCO.1.2E**

2905 **9.9.1.4.1 Work unit ACE_CCO.1-8**

2906 The evaluator shall check that the PP-Configuration made up of all the Protection Profiles and
2907 PP-Modules identified in the components statement of the PP-Configuration is consistent.
2908 That is, the evaluator shall check that no contradiction arises from the whole set of Protection
2909 Profiles and PP-Modules included in the PP-Configuration.

2910 The evaluator can organise this work in many ways; the actual organisation may depend on
2911 the will to derive evaluation results for more than one PP-Configuration at a time

2912 For instance, the evaluator can process in two steps as follows:

- 2913 a) Assess the consistency of the set of Protection Profiles composing the PP-Configuration,
- 2914 b) Then proceed with the assessment of the PP-Configuration consistency incrementally, by
2915 adding one PP-Module at a time.

2916 An alternative is to proceed incrementally but mixing PPs and PP-Modules or to
2917 flatten/serialise the definition of the PP-Configuration (cf. Annex B in ISO/IEC 15408-1),
2918 duplicating as required, and to assess the consistency of the whole set of elements.

2919 Any incremental consistency analysis step where C is a subset of the PP-Configuration and X is
2920 a PP or a PP-Module that has to be added to C consists in:

- 2921 • assessing that the SPD, the objectives and the SFRs of X do not contradict the statements in
2922 C;
- 2923 • the assumptions and objectives for the environment in X either are the same as in C or
2924 address security aspects that are out of the scope of C.

2925 If the PP-Configuration is a Direct Rationale PP-Configuration (as determined in ACE_CCO.1-2)
2926 then the TOE objectives are not required in the consistency analysis.

2927 Note that if X is a PP-Module, C contains all its Base-PP(s) and Evaluation of sub-activity
2928 (ACE_MCO.1) has succeed for X, then the consistency analysis step has to be performed with
2929 respect to the components of C different from these Base-PP(s) only.

2930 **10 Class ASE: Security Target evaluation**

2931 **10.1 Introduction**

2932 This Clause describes the evaluation of an ST. The ST evaluation should be started prior to any TOE
2933 evaluation sub-activities since the ST provides the basis and context to perform these sub-activities.
2934 The evaluation methodology in this subclause is based on the requirements on the ST as specified
2935 in ISO/IEC 15408-3 class ASE.

2936 This Clause should be used in conjunction with Annexes A, B and C, **Guidance for Operations** in
2937 ISO/IEC 15408-1, as these Annexes clarify the concepts here and provide many examples.

2938 **10.2 Application notes**

2939 **10.2.1 Re-using the evaluation results of certified PPs**

2940 While evaluating an ST that is based on one or more certified PPs, it may be possible to re-use the
2941 fact that these PPs were certified. The potential for re-use of the result of a certified PP is greater if
2942 the ST does not add threats, OSPs, assumptions, security objectives and/or security requirements
2943 to those of the PP. If the ST contains much more than the certified PP, re-use may not be useful at
2944 all.

2945 The evaluator is allowed to re-use the PP evaluation results by doing certain analyses only partially
2946 or not at all if these analyses or parts thereof were already done as part of the PP evaluation. While
2947 doing this, the evaluator should assume that the analyses in the PP were performed correctly.

2948 An example would be where the PP contains a set of security requirements, and these were
2949 determined to be internally consistent during the PP evaluation. If the ST uses the exact same
2950 requirements, the consistency analysis does not have to be repeated during the ST evaluation. If
2951 the ST adds one or more requirements, or performs operations on these requirements, the analysis
2952 will have to be repeated. However, it may be possible to save work in this consistency analysis by
2953 using the fact that the original requirements are internally consistent. If the original requirements
2954 are internally consistent, the evaluator only has to determine that:

- 2955 a) the set of all new and/or changed requirements is internally consistent, and
- 2956 b) the set of all new and/or changed requirements is consistent with the original
2957 requirements.

2958 The evaluator notes in the ETR each case where analyses are not done or only partially done for
2959 this reason.

2960 The same re-use discussion applies to an ST claiming conformance to a certified PP-Configuration.

2961 **10.3 ST introduction (ASE_INT)**

2962 **10.3.1 Evaluation of sub-activity (ASE_INT.1)**

2963 **10.3.1.1 Objectives**

2964 The objective of this sub-activity is to determine whether the ST and the TOE are correctly
2965 identified, whether the TOE is correctly described in a narrative way at three levels of abstraction
2966 (TOE reference, TOE overview and TOE description), and whether these three descriptions are
2967 consistent with each other.

2968 **10.3.1.2 Input**

2969 The evaluation evidence for this sub-activity is:

2970 a) the ST.

2971 **10.3.1.3 Action ASE_INT.1.1E**

2972 ISO/IEC 15408-3 ASE_INT.1.1C: *The ST introduction shall contain an ST reference, a TOE reference, a*
2973 *TOE overview and a TOE description.*

2974 **10.3.1.3.1 Work unit ASE_INT.1-1**

2975 The evaluator **shall check** that the ST introduction contains an ST reference, a TOE reference, a
2976 TOE overview and a TOE description.

2977 ISO/IEC 15408-3 ASE_INT.1.2C: *The ST reference shall uniquely identify the ST.*

2978 **10.3.1.3.2 Work unit ASE_INT.1-2**

2979 The evaluator **shall examine** the ST reference to determine that it uniquely identifies the ST.

2980 The evaluator determines that the ST reference identifies the ST itself, so that it may be easily
2981 distinguished from other STs, and that it also uniquely identifies each version of the ST, e.g. by
2982 including a version number and/or a date of publication.

2983 In evaluations where a CM system is provided, the evaluator may validate the uniqueness of the
2984 reference by checking the configuration list. In the other cases, the ST should have some
2985 referencing system that is capable of supporting unique references (e.g. use of numbers, letters or
2986 dates).

2987 ISO/IEC 15408-3 ASE_INT.1.3C: *The TOE reference shall identify the TOE.*

2988 **10.3.1.3.3 Work unit ASE_INT.1-3**

2989 The evaluator **shall examine** the TOE reference to determine that it uniquely identifies the TOE.

2990 The evaluator determines that the TOE reference uniquely identifies the TOE, so that it is clear to
2991 which TOE the ST refers, and that it also identifies the version of the TOE, e.g. by including a
2992 version/release/build number, or a date of release.

2993 In the end of the evaluation, the evaluator **shall check** the TOE reference, and any unique
2994 identifiers associated with the TOE physical components are consistent with the identifier(s)
2995 assigned to the TOE evaluated in work units related to ALC_CMC.x.1C and the configuration list
2996 evaluated in work units related to ALC_CMS.x.2C.

- 2997 **10.3.1.3.4 Work unit ASE_INT.1-4**
- 2998 The evaluator ***shall examine*** the TOE reference to determine that it is not misleading.
- 2999 If the TOE is related to one or more well-known products, it is allowed to reflect this in the TOE
3000 reference. However, this should not be used to mislead consumers and it must be made clear which
3001 part of the product has been evaluated.
- 3002 When a TOE needs some required non-TOE hardware/software/firmware to run properly, the TOE
3003 reference may include the name of the non-TOE hardware/software/firmware used by the TOE,
3004 however it must be made clear that the non-TOE hardware/software/firmware has not been
3005 evaluated.
- 3006 ISO/IEC 15408-3 ASE_INT.1.4C: *The TOE overview shall summarise the usage and major security*
3007 *features of the TOE.*
- 3008 **10.3.1.3.5 Work unit ASE_INT.1-5**
- 3009 The evaluator ***shall examine*** the TOE overview to determine that it describes the usage and major
3010 security features of the TOE.
- 3011 The TOE overview may describe security features that are provided by the product, and/or those
3012 that users may expect in that product type, but it must clearly distinguish those features that are
3013 evaluated and those that are not evaluated.
- 3014 The TOE overview shall be consistent with information provided in other sections of the Security
3015 Target such as the TOE description, the security objectives, the security functional requirements,
3016 and the TOE summary specification. In addition to ensuring the evaluated security features are
3017 consistently described throughout the ST, this means that any security feature that is not evaluated
3018 is only discussed within the ST introduction, or else is explicitly identified as not evaluated in each
3019 other place where it is mentioned (failure to make this identification means that this work unit is
3020 assigned a fail verdict).
- 3021 The TOE overview in an ST for a composed TOE should describe the usage and major security
3022 feature of the composed TOE, rather than those of the individual component TOEs.
- 3023 The evaluator determines that the overview is clear enough for consumers, and sufficient to give
3024 them a general understanding of the intended usage and major security features of the TOE.
- 3025 ISO/IEC 15408-3 ASE_INT.1.5C: *The TOE overview shall identify the TOE type.*
- 3026 **10.3.1.3.6 Work unit ASE_INT.1-6**
- 3027 The evaluator ***shall check*** that the TOE overview identifies the TOE type.
- 3028 **10.3.1.3.7 Work unit ASE_INT.1-7**
- 3029 The evaluator ***shall examine*** the TOE overview to determine that the TOE type is not misleading.
- 3030 There are situations where the general consumer would expect certain functionality of the TOE
3031 because of its TOE type. If this functionality is absent in the TOE, the evaluator determines that the
3032 TOE overview adequately discusses this absence.
- 3033 There are also TOEs where the general consumer would expect that the TOE should be able to
3034 operate in a certain operational environment because of its TOE type. If the TOE is unable to
3035 operate in such an operational environment, the evaluator determines that the TOE overview
3036 adequately discusses this.

3037 ISO/IEC 15408-3 ASE_INT.1.6C: *The TOE overview shall identify any non-TOE*
3038 *hardware/software/firmware required by the TOE.*

3039 **10.3.1.3.8 Work unit ASE_INT.1-8**

3040 The evaluator ***shall examine*** the TOE overview to determine that it identifies any non-TOE
3041 hardware/software/firmware required by the TOE.

3042 While some TOEs are able to run stand-alone, other TOEs (notably software TOEs) need additional
3043 hardware, software or firmware to operate. If the TOE does not require any hardware, software or
3044 firmware, this work unit is not applicable and therefore considered to be satisfied.

3045 The evaluator determines that the TOE overview identifies any additional hardware, software and
3046 firmware needed by the TOE to operate. This identification does not have to be exhaustive, but
3047 detailed enough for potential consumers of the TOE to determine whether their current hardware,
3048 software and firmware support use of the TOE, and, if this is not the case, which additional
3049 hardware, software and/or firmware is needed.

3050 ISO/IEC 15408-3 ASE_INT.1.7C: *The TOE description shall describe the physical scope of the TOE.*

3051 **10.3.1.3.9 Work unit ASE_INT.1-9**

3052 The evaluator ***shall examine*** the TOE description to determine that it describes the physical scope
3053 of the TOE.

3054 The evaluator determines that the TOE description lists the hardware, firmware, software and
3055 guidance parts that constitute the TOE and describes them at a level of detail that is sufficient to
3056 give the reader a general understanding of those parts.

3057 As a minimum, the TOE description will cover the following elements:

3058 a) Each separately delivered part of the TOE, which will be identified by its unique identifier
3059 and the current format (binary, wafer, inlay, *.pdf, *.doc, *.chm etc.).

3060 b) The delivery method used by the developer to make available each part to the TOE
3061 consumer (Web site download, courier delivery, etc.)

3062 The physical description will also include some clear statements about the evaluated TOE
3063 configuration. In the case where a product could have multiple physical components, and therefore
3064 multiple configurations, the evaluated configurations must be briefly described and identified.

3065 The evaluator also determines that there is no possible misunderstanding as to whether any
3066 hardware, firmware, software or guidance part is part of the TOE or not.

3067 ISO/IEC 15408-3 ASE_INT.1.8C: *The TOE description shall describe the logical scope of the TOE.*

3068 **10.3.1.3.10 Work unit ASE_INT.1-10**

3069 The evaluator ***shall examine*** the TOE description to determine that it describes the logical scope of
3070 the TOE.

3071 The evaluator determines that the TOE description discusses the logical security features offered
3072 by the TOE at a level of detail that is sufficient to give the reader a general understanding of those
3073 features.

3074 The evaluator also determines that there is no possible misunderstanding as to whether any logical
3075 security feature is offered by the TOE or not.

3076 An ST for a composed TOE may refer out to the description of the logical scope of the component
 3077 TOEs, provided in the component TOE STs to provide the majority of this description for the
 3078 composed TOE. However, the evaluator determines that the composed TOE ST clearly discusses
 3079 which features of the individual components are not within the composed TOE, and therefore not a
 3080 feature of the composed TOE.

3081 **10.3.1.4 Action ASE_INT.1.2E**

3082 **10.3.1.4.1 Work unit ASE_INT.1-11**

3083 The evaluator *shall examine* the TOE reference, TOE overview and TOE description to determine
 3084 that they are consistent with each other.

3085 **10.4 Conformance claims (ASE_CCL)**

3086 **10.4.1 Evaluation of sub-activity (ASE_CCL.1)**

3087 **10.4.1.1 Objectives**

3088 The objective of this sub-activity is to determine the validity of various conformance claims. These
 3089 describe how the ST and the TOE conform to ISO/IEC 15408 and how the ST conforms to a PP-
 3090 Configuration, PPs and packages.

3091 **10.4.1.2 Input**

3092 The evaluation evidence for this sub-activity is:

- 3093 a) the ST;
- 3094 b) the Base-PP(s) that the ST claims conformance to;
- 3095 c) the package(s) that the ST claims conformance to.

3096 **10.4.1.3 Action ASE_CCL.1.1E**

3097 ISO/IEC 15408-3 ASE_CCL.1.1C: *The conformance claim shall contain an ISO/IEC 15408*
 3098 *conformance claim that identifies the version of ISO/IEC 15408 to which the ST and the TOE claim*
 3099 *conformance.*

3100 **10.4.1.3.1 Work unit ASE_CCL.1-1**

3101 The evaluator *shall check* that the conformance claim contains an ISO/IEC 15408 conformance
 3102 claim that identifies the version of ISO/IEC 15408 to which the ST and the TOE claim conformance.

3103 The evaluator determines that ISO/IEC 15408 conformance claim identifies the version of ISO/IEC
 3104 15408 that was used to develop this ST. This should include the version number of ISO/IEC 15408
 3105 and, unless the International English version of ISO/IEC 15408 was used, the language of the
 3106 version of ISO/IEC 15408 that was used.

3107 For a composed TOE, the evaluator will consider any differences between the version of ISO/IEC
 3108 15408 claimed for a component and the version of ISO/IEC 15408 claimed for the composed TOE.
 3109 If the versions differ the evaluator will assess whether the differences between the versions will
 3110 lead to conflicting claims.

3111 For instances where ISO/IEC 15408 conformance claims for the base TOE and dependent TOE are
 3112 for different major releases of ISO/IEC 15408 (e.g. one component TOE conformance claim is
 3113 ISO/IEC 15408 v2.x and the other component TOE conformance claim is ISO/IEC 15408 v3.x), the
 3114 conformance claim for the composed TOE will be the earlier release of ISO/IEC 15408, as ISO/IEC

3115 15408 is developed with an aim to provide backwards compatibility (although this may not be
3116 achieved in the strictest sense, it is understood to be achieved in principle).

3117 ISO/IEC 15408-3 ASE_CCL.1.2C: *ISO/IEC 15408 conformance claim shall describe the conformance of*
3118 *the ST to ISO/IEC 15408-2 as either ISO/IEC 15408-2 conformant or ISO/IEC 15408-2 extended.*

3119 **10.4.1.3.2 Work unit ASE_CCL.1-2**

3120 The evaluator **shall check** that ISO/IEC 15408 conformance claim states a claim of either ISO/IEC
3121 15408-2 conformant or ISO/IEC 15408-2 extended for the ST.

3122 For a composed TOE, the evaluator will consider whether this claim is consistent not only with
3123 ISO/IEC 15408-2, but also with the claims of conformance to ISO/IEC 15408-2 by each of the
3124 component TOEs. I.e. if one or more component TOEs claims to be ISO/IEC 15408-2 extended, then
3125 the composed TOE should also claim to be ISO/IEC 15408-2 extended.

3126 ISO/IEC 15408 conformance claim for the composed TOE may be ISO/IEC 15408-2 extended, even
3127 though the component TOEs are ISO/IEC 15408-2 conformant, in the event that additional SFRs
3128 are claimed for the base TOE (see composed TOE guidance for ASE_CCL.1.6C)

3129 ISO/IEC 15408-3 ASE_CCL.1.3C: *ISO/IEC 15408 conformance claim shall describe the conformance of*
3130 *the ST to ISO/IEC 15408-3 as either ISO/IEC 15408-3 conformant or ISO/IEC 15408-3 extended.*

3131 **10.4.1.3.3 Work unit ASE_CCL.1-3**

3132 The evaluator **shall check** that ISO/IEC 15408 conformance claim states a claim of either ISO/IEC
3133 15408-3 conformant or ISO/IEC 15408-3 extended for the ST.

3134 ISO/IEC 15408-3 ASE_CCL.1.4C: *ISO/IEC 15408 conformance claim shall be consistent with the*
3135 *extended components definition.*

3136 **10.4.1.3.4 Work unit ASE_CCL.1-4**

3137 The evaluator **shall examine** ISO/IEC 15408 conformance claim for ISO/IEC 15408-2 to determine
3138 that it is consistent with the extended components definition.

3139 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-2 conformant, the evaluator
3140 determines that the extended components definition does not define functional components.

3141 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-2 extended, the evaluator determines
3142 that the extended components definition defines at least one extended functional component.

3143 **10.4.1.3.5 Work unit ASE_CCL.1-5**

3144 The evaluator **shall examine** ISO/IEC 15408 conformance claim for ISO/IEC 15408-3 to determine
3145 that it is consistent with the extended components definition.

3146 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-3 conformant, the evaluator
3147 determines that the extended components definition does not define assurance components.

3148 If ISO/IEC 15408 conformance claim contains ISO/IEC 15408-3 extended, the evaluator determines
3149 that the extended components definition defines at least one extended assurance component.

3150 ISO/IEC 15408-3 ASE_CCL.1.5C: *The conformance claim shall identify a PP-Configuration, or all PPs*
3151 *and security requirement packages to which the ST claims conformance.*

3152 **10.4.1.3.6 Work unit ASE_CCL.1-6**

3153 The evaluator ***shall check*** that the conformance claim contains a PP claim that identifies all PPs for
3154 which the ST claims conformance.

3155 If the ST does not claim conformance to a PP, this work unit is not applicable and therefore
3156 considered to be satisfied.

3157 The evaluator determines that any referenced PPs are unambiguously identified (e.g. by title and
3158 version number, or by the identification included in the introduction of that PP).

3159 For conformance claims to PPs containing functional packages, the evaluator examines that:

- 3160 - all mandatory packages from the PP have been selected into the ST;
- 3161 - packages that the PP has declared as mandatory if the TOE meets certain conditions have
3162 been included if that condition is met by the TOE.
- 3163 - all dependencies between the selected packages have been resolved.

3164 The evaluator is reminded that claims of partial conformance to a PP are not permitted. Therefore,
3165 conformance to a PP requiring a composite solution may be claimed in an ST for a composed TOE.
3166 Conformance to such a PP would not have been possible during the evaluation of the component
3167 TOEs, as these components would not have satisfied the composed solution. This is only possible in
3168 the instances where the “composite” PP permits use of the composition evaluation approach (use
3169 of ACO components).

3170 For PPs containing functional packages, partial conformance means that not all mandatory
3171 packages have been included in the ST, a mandatory or optional functional package has only been
3172 partially included into the ST, or a dependency requirement between functional packages has not
3173 been met.

3174 **10.4.1.3.7 Work unit ASE_CCL.1-6a**

3175 The evaluator ***shall check*** that, for each PP to which the ST claims conformance, the conformance
3176 statement of that PP allows all other PPs in the conformance claim to be allowed to be claimed with
3177 that PP.

3178 If the ST does not claim conformance to a PP, or claims conformance to only one PP, this work unit
3179 is not applicable and therefore considered to be satisfied.

3180 If the ST is not claiming exact conformance to a PP, this work unit is not applicable and therefore
3181 considered to be satisfied.

3182 The evaluator determines that the conformance statement of the PP to which conformance is being
3183 claimed lists each of the PPs identified in the conformance claim section of the ST as being “allowed
3184 to be claimed with” that PP. Note that this is only applicable in cases where that PP requires exact
3185 conformance and the ST claims exact conformance.

3186 EXAMPLE consider the case where an ST is being evaluated and claims conformance to PPs B and
3187 C; this is depicted in Figure 7. The ST is claiming exact conformance, so all PPs require exact
3188 conformance in their conformance statements. Under this work unit, the evaluator determines that
3189 PP B lists (in its conformance statement) “PP C” as being a PP that can be claimed (by an ST) with
3190 PP B. Likewise, the evaluator determines that PP C lists (in its conformance statement) “PP B” as
3191 being a PP that can be claimed (by an ST) with PP C.

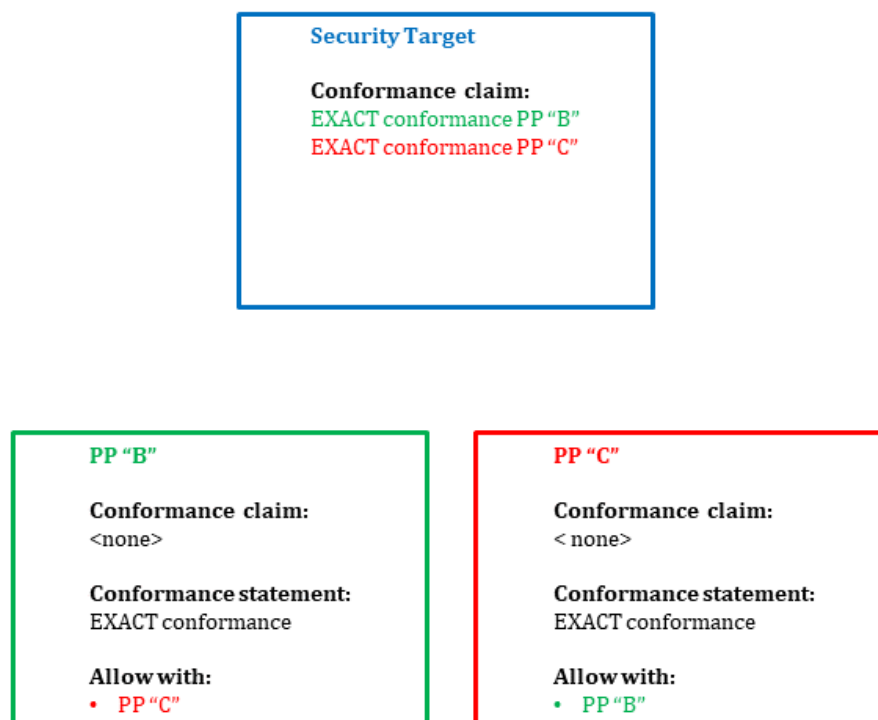


Figure 7 — Example of exact conformance relationships between an ST and PPs

10.4.1.3.8 Work unit ASE_CCL.1-6b

The evaluator shall check that the conformance claim contains a PP-Configuration claim that identifies the PP-Configuration(s) for which the ST claims conformance.

If the ST does not claim conformance to a PP-Configuration, this work unit is not applicable and therefore considered to be satisfied.

If the ST claims conformance to multiple PP-Configurations, the evaluator ensures that the conformance statement for all PP-Configurations is either "strict" or "demonstrable"; an ST cannot claim exact conformance to multiple PP-Configurations

If the ST claims conformance to a PP-Configuration and a PP (that is not part of the PP-Configuration), the evaluator ensures that the conformance statement for all PP-Configurations and the PP is either "strict" or "demonstrable"; an ST cannot claim exact conformance to a PP-Configurations and a PP that is not part of the PP-Configuration. The evaluator determines that any referenced PP-Configuration(s) are unambiguously identified (e.g. by title and version number, or by the identification included in the introduction of that PP).

For conformance claims to PP-Configurations containing functional packages, the evaluator examines that:

- all mandatory packages from the PP-Configuration components have been selected into the ST;
- packages that a PP-Configuration component has declared as mandatory if the TOE meets certain conditions have been included if that condition is met by the TOE;
- all dependencies between the selected packages have been resolved.

- 3215 The evaluator is reminded that claims of partial conformance to a PP are not permitted. For PP-
 3216 Configurations containing functional packages, partial conformance means that not all mandatory
 3217 packages have been included in the ST, a mandatory or optional functional package has only been
 3218 partially included into the ST, or a dependency requirement between functional packages has not
 3219 been met.
- 3220 **10.4.1.3.9 Work unit ASE_CCL.1-7**
- 3221 The evaluator **shall check** that the conformance claim contains a package claim that identifies all
 3222 packages to which the ST claims conformance.
- 3223 If the ST does not claim conformance to a package, this work unit is not applicable and therefore
 3224 considered to be satisfied.
- 3225 The evaluator determines that any referenced packages are unambiguously identified (e.g. by title
 3226 and version number, or by the identification included in the introduction of that package). The
 3227 evaluator determines that if the ST is claiming exact conformance to PPs or a PP-Configuration,
 3228 then only packages specified in the PP(s), Base-PP(s), or the PP-Module(s) are included in the ST's
 3229 package claim.
- 3230 The evaluator determines that the component TOE STs from which the composed TOE is derived
 3231 are also unambiguously identified.
- 3232 The evaluator is reminded that claims of partial conformance to a package are not permitted.
- 3233 **10.4.1.3.10 Work unit ASE_CCL.1-8**
- 3234 The evaluator **shall check** that, for each identified package, the conformance claim states a claim of
 3235 either package-name conformant or package-name augmented.
- 3236 If the ST claims conformance to a PP and the PP itself claims conformance to one or more
 3237 functional packages then the ST shall not separately make a conformance claim to the same
 3238 packages in the same mode (conformant or augmented). However, if an ST makes an augmentation
 3239 to the version of the package to which the PP conforms, then the ST shall separately make this
 3240 package-augmented conformance claim. Where an ST claims exact conformance to the PP, it shall
 3241 adopt only the form of the package to which the PP conforms, and therefore shall not make
 3242 separate claims of this sort.
- 3243 If the ST does not claim conformance to a package, this work unit is not applicable and therefore
 3244 considered to be satisfied.
- 3245 If the package conformance claim contains package-name conformant, the evaluator determines
 3246 that:
- 3247 a) If the package is an assurance package, then the ST contains all SARs included in the
 3248 package, but no additional SARs.
- 3249 b) If the package is a functional package, then all assumptions, threats, OSPs, security
 3250 objectives and SFRs included in the package are identical to those included in the ST
 3251 (after allowing any remaining assignments or selections from the package to be made in
 3252 the ST).
- 3253 If the package conformance claim contains package-name augmented, the evaluator determines
 3254 that:
- 3255 a) If the package is an assurance package then the ST contains all SARs included in the
 3256 package, and at least one additional SAR or at least one SAR that is hierarchical to a SAR
 3257 in the package.

3258 If the package is a functional package, then the constituent parts (security problem definition,
3259 security objectives, SFRs) of that ST contain all constituent parts (security problem definition,
3260 security objectives, SFRs) of that specific package, but additionally contain at least one
3261 enhancement of the security
3262 functionality defined by that specific package (finally resulting in an additional SFR or one an SFR
3263 that is hierarchically higher than an SFR in the package).

3264 The evaluator determines that, if the ST claims exact conformance to the PPs/PP-Configuration,
3265 only claims of <package name>-conformant are present.

3266 ISO/IEC 15408-3 ASE_CCL.1.7C: *The conformance claim rationale shall demonstrate that the TOE*
3267 *type is consistent with the TOE type in the PP-Configuration or PPs for which conformance is being*
3268 *claimed.*

3269 **10.4.1.3.11 Work unit ASE_CCL.1-9**

3270 In this work unit, the term “PP” shall be understood to mean “PP or PP-Configuration component”.

3271 The evaluator ***shall examine*** the conformance claim rationale to determine that the TOE type of
3272 the TOE is consistent with all TOE types of the PPs.

3273 If the ST does not claim conformance to a PP, this work unit is not applicable and therefore
3274 considered to be satisfied.

3275 The relation between the types may be simple: a firewall ST claiming conformance to a firewall PP,
3276 or more complex: a smart card ST claiming conformance to a number of PPs at the same time (a PP
3277 for the integrated circuit, a PP for the smart card OS, and two PPs for two applications on the smart
3278 card).

3279 For a composed TOE, the evaluator will determine whether the conformance claim rationale
3280 demonstrates that the TOE types of the component TOEs are consistent with the composed TOE
3281 type. This does not mean that both the component and the composed TOE types have to be the
3282 same, but rather that the component TOEs are suitable for integration to provide the composed
3283 TOE. It should be made clear in the composed TOE ST which SFRs are only included as a result of
3284 composition, and were not examined as SFRs in the base and dependent TOE (e.g. EALx) evaluation.

3285 ISO/IEC 15408-3 ASE_CCL.1.8C: *The conformance claim rationale shall demonstrate that the*
3286 *statement of the security problem definition is consistent with the statement of the security problem*
3287 *definition in the PP-Configuration or PPs for which conformance is being claimed.*

3288 **10.4.1.3.12 Work unit ASE_CCL.1-10**

3289 In this work unit, the term “PP” shall be understood to mean “PP or PP-Configuration component”.

3290 The evaluator ***shall examine*** the conformance claim rationale to determine that it demonstrates
3291 that the statement of security problem definition is consistent, as defined by the conformance
3292 statement of the PP, with the statements of security problem definition stated in the PPs to which
3293 conformance is being claimed.

3294 If the ST does not claim conformance with a PP, this work unit is not applicable and therefore
3295 considered to be satisfied.

3296 If the PP does not have a statement of security problem definition, this work unit is not applicable
3297 and therefore considered to be satisfied.

3298 If the PP contains functional packages, the evaluator determines that the security problem
3299 definition of the ST consists of all assumptions, threats and OSPs of all mandatory and all selected
3300 optional functional packages.

- 3301 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
3302 demonstrable conformance also hold for the SPD descriptions taken from the packages.
- 3303 If exact conformance is required by the PP to which conformance is being claimed, no conformance
3304 claim rationale is required. Instead, the evaluator determines whether:
- 3305 a) the threats in the ST are identical (no fewer threats, no additional threats) to the threats in
3306 the PP to which conformance is being claimed. If exact conformance is being claimed to
3307 more than one PP, then the set of threats in the ST must be identical to the union of the
3308 threats in all PPs to which conformance is being claimed.
 - 3309 b) the OSPs in the ST are identical (no fewer OSPs, no additional OSPs) to the OSPs in the PP
3310 to which conformance is being claimed. If exact conformance is being claimed to more than
3311 one PP, then the set of OSPs in the ST must be identical to the union of the OSPs in all PPs
3312 to which conformance is being claimed.
 - 3313 c) the assumptions in the ST are identical (no fewer assumptions, no additional assumptions)
3314 to the assumptions in the PP to which conformance is being claimed. If exact conformance
3315 is being claimed to more than one PP, then the set of assumptions in the ST must be
3316 identical to the union of the assumptions in all PPs to which conformance is being claimed,
3317 with the following possible exception;
 - 3318 - an assumption (or part of an assumption) from a PP can be omitted, if all security
3319 objectives for the operational environment addressing this assumption (or part of an
3320 assumption) are replaced by security objectives for the TOE that are identical to
3321 (taken from) another of the PPs to which the ST is claiming conformance;
- 3322 When examining an ST in these circumstances (assumptions from one PP are replaced by security
3323 objectives on the TOE from one of the other PPs) the evaluator shall carefully determine that the
3324 condition given above is fulfilled. The following discussion gives an example:
- 3325 - EXAMPLE an ST is claiming exact conformance to two PPs. As determined in
3326 previous work units, both PPs require exact conformance in their conformance
3327 statements, and both PPs list the other as being "allowed with" the PP in a
3328 conformance claim by an ST. One PP to which the ST claims conformance contains an
3329 assumption stating that the operational environment prevents unauthorised
3330 modification or interception of data sent to an external interface of the TOE. This may
3331 be the case if the TOE accepts data in clear text and without integrity protection at
3332 this interface and is assumed to be located in a secure operational environment,
3333 which will prevent attackers from accessing this data. The assumption will then be
3334 mapped in the PP to some objective for the operational environment stating that the
3335 data interchanged at this interface are protected by adequate measures in the
3336 operational environment. Suppose there is another PP that specifies that conformant
3337 TOEs must protect data sent over the TOEs external interfaces, and has appropriate
3338 threats and security objectives addressing this threat. The ST author can then
3339 replace the assumption and security objective for the environment related to the
3340 protection of data over the external interfaces of the TOE from one PP with the
3341 security objective stating that the TOE itself protects these data, for example by
3342 providing a secure channel for encryption and integrity protection of all data
3343 transferred via this interface from the other PP; the corresponding objective and
3344 assumption for the operational environment from the other PP is thus omitted from
3345 the ST. This is also called re-assigning of the objective, since the objective is re-
3346 assigned from the operational environment to the TOE. Note, that this TOE is still
3347 secure in an operational environment fulfilling the omitted assumption and therefore
3348 still fulfils the PP. Further, the set of threats and objectives in the ST is still no
3349 broader than the union of threats and objectives in the PPs to which it is claiming
3350 exact conformance.

3351 If strict conformance is required by the PP to which conformance is being claimed no conformance
3352 claim rationale is required. Instead, the evaluator determines whether:

- 3353 a) the threats in the ST are a superset of or identical to the threats in the PP to which
3354 conformance is being claimed;
- 3355 b) the OSPs in the ST are a superset of or identical to the OSPs in the PP to which
3356 conformance is being claimed;
- 3357 c) the assumptions in the ST are identical to the assumptions in the PP to which
3358 conformance is being claimed, with two possible exceptions described in the following
3359 two bullet points;
- 3360 — an assumption (or part of an assumption) from the PP can be omitted, if all security objectives
3361 for the operational environment addressing this assumption (or part of an assumption) are
3362 replaced by security objectives for the TOE;
- 3363 — an assumption can be added to the assumptions defined in the PP, if a rationale is given, why
3364 the new assumption neither mitigates a threat (or a part of a threat) meant to be addressed by
3365 security objectives for the TOE in the PP, nor fulfils an OSP (or part of an OSP) meant to be
3366 addressed by security objectives for the TOE in the PP.

3367 When examining an ST claiming a PP, which omits assumptions from the PP or adds new
3368 assumptions, the evaluator shall carefully determine, if the conditions given above are fulfilled. The
3369 following discussion gives some motivation and examples for these cases:

3370 — Example for omitting an assumption: A PP may contain an assumption stating that the
3371 operational environment prevents unauthorised modification or interception of data sent to an
3372 external interface of the TOE. This may be the case if the TOE accepts data in clear text and
3373 without integrity protection at this interface and is assumed to be located in a secure
3374 operational environment, which will prevent attackers from accessing these data. The
3375 assumption will then be mapped in the PP to some objective for the operational environment
3376 stating that the data interchanged at this interface are protected by adequate measures in the
3377 operational environment. If an ST claiming this PP defines a more secure TOE, which has an
3378 additional security objective stating that the TOE itself protects these data, for example by
3379 providing a secure channel for encryption and integrity protection of all data transferred via
3380 this interface, the corresponding objective and assumption for the operational environment
3381 can be omitted from the ST. This is also called re-assigning of the objective, since the objective
3382 is re-assigned from the operational environment to the TOE. Note, that this TOE is still secure
3383 in an operational environment fulfilling the omitted assumption and therefore still fulfils the
3384 PP.

3385 — Example for adding an assumption: In this example, the PP is designed to specify requirements
3386 for a TOE of type "Firewall" and an ST author wishes to claim this PP for a TOE, which
3387 implements a firewall, but additionally provides the functionality of a virtual private network
3388 (VPN) component. For the VPN functionality, the TOE needs cryptographic keys and these keys
3389 may also have to be handled securely by the operational environment (e. g. if symmetric keys
3390 are used to secure the network connection and therefore need to be provided in some secure
3391 way to other components in the network). In this case, it is acceptable to add an assumption
3392 that the cryptographic keys used by the VPN are handled securely by the operational
3393 environment. This assumption does not address threats or OSPs of the PP and therefore fulfils
3394 the conditions stated above.

3395 — Counterexample for adding an assumption: In a variant of the first example a PP may already
3396 contain an objective for the TOE to provide a secure channel for one of its interfaces, and this
3397 objective is mapped to a threat of unauthorised modification or reading of the data on this
3398 interface. In this case, it is clearly not allowed for an ST claiming this PP to add an assumption
3399 for the operational environment, which assumes that the operational environment protects

- 3400 data on this interface against modification or unauthorised reading of the data. This
 3401 assumption would reduce a threat, which is meant to be addressed by the TOE. Therefore a
 3402 TOE fulfilling an ST with this added assumption would not automatically fulfil the PP any more
 3403 and this addition is therefore not allowed.
- 3404 — Second counterexample for adding an assumption: In the example above of a TOE
 3405 implementing a firewall it would not be admissible to add a general assumption that the TOE is
 3406 only connected to trusted devices, because this would obviously remove essential threats
 3407 relevant for a firewall (namely that there is untrusted IP traffic, which needs to be filtered).
 3408 Therefore, this addition would not be allowed.
- 3409 If demonstrable conformance is required by the PP, the evaluator examines the conformance claim
 3410 rationale to determine that it demonstrates that the statement of security problem definition of the
 3411 ST is equivalent or more restrictive than the statement of security problem definition in the PP to
 3412 which conformance is being claimed.
- 3413 For this, the conformance claim rationale needs to demonstrate that the security problem
 3414 definition in the ST is equivalent (or more restrictive) than the security problem definition in the
 3415 PP. This means that:
- 3416 — all TOEs that would meet the security problem definition in the ST also meet the security
 3417 problem definition in the PP. This can also be shown indirectly by demonstrating that every
 3418 event, which realises a threat defined in the PP or violates an OSP defined in the PP, would also
 3419 realise a threat stated in the ST or violate an OSP defined in the ST. Note that fulfilling an OSP
 3420 stated in the ST may avert a threat stated in the PP or that averting a threat stated in the ST
 3421 may fulfil an OSP stated in the PP, so threats and OSPs can substitute each other;
- 3422 — all operational environments that would meet the security problem definition in the PP would
 3423 also meet the security problem definition in the ST (with one exception in the next bullet);
- 3424 — besides a set of assumptions in the ST needed to demonstrate conformance to the SPD of the
 3425 PP, an ST may specify further assumptions, but only if these additional assumptions are
 3426 independent of and do not affect the security problem definition as defined in the PP. More
 3427 detailed, there are no assumptions in the ST that exclude threats to the TOE that need to be
 3428 countered by the TOE according to the PP. Similarly, there are no assumptions in the ST that
 3429 realise aspects of an OSP stated in the PP, which are meant to be fulfilled by the TOE according
 3430 to the PP."
- 3431 For a composed TOE, the evaluator will consider whether the security problem definition of the
 3432 composed TOE is consistent with that specified in the STs for the component TOEs. This is
 3433 determined in terms of demonstrable conformance. In particular, the evaluator examines the
 3434 conformance claim rationale to determine that:
- 3435 a) Threat statements and OSPs in the composed TOE ST do not contradict those from the
 3436 component STs.
- 3437 b) Any assumptions made in the component STs are upheld in the composed TOE ST. That is,
 3438 either the assumption should also be present in the composed ST, or the assumption
 3439 should be positively addressed in the composed ST. The assumption may be positively
 3440 addressed through specification of requirements in the composed TOE to provide
 3441 functionality fulfilling the concern captured in the assumption.
- 3442 ISO/IEC 15408-3 ASE_CCL.1.9C: *The conformance claim rationale shall demonstrate that the*
 3443 *statement of security objectives is consistent with the statement of security objectives in the PP-*
 3444 *Configuration or PPs for which conformance is being claimed.*

3445 **10.4.1.3.13 Work unit ASE_CCL.1-11**

3446 In this work unit, the term “PP” shall be understood to mean “PP or PP-Configuration component”.

3447 The evaluator *shall examine* the conformance claim rationale to determine that the statement of
3448 security objectives is consistent, as defined by the conformance statement of the PP, with the
3449 statement of security objectives in the PPs to which conformance is being claimed.

3450 If the ST does not claim conformance to a PP, this work unit is not applicable and therefore
3451 considered to be satisfied.

3452 If the PP to which conformance is being claimed contains functional packages, the evaluator
3453 determines that the security objectives of the ST consist of all security objectives of all mandatory
3454 and all selected optional functional packages.

3455 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
3456 demonstrable conformance also hold for the security objectives taken from the packages.

3457 If exact conformance is required by the PP to which conformance is being claimed, no conformance
3458 claim rationale is required. Instead, the evaluator determines whether:

3459 a) The ST contains all security objectives for the TOE of the PP to which conformance is being
3460 claimed. Note that in the exact conformance case, it is not allowed for the ST under
3461 evaluation to have additional security objectives for the TOE. If conformance is being
3462 claimed to more than one PP, the set of security objectives for the TOE must be identical to
3463 the union of the security objectives for the TOE in the PPs to which conformance is being
3464 claimed.

3465 b) The security objectives for the operational environment in the ST are identical to the
3466 security objectives for the operational environment in the PP to which conformance is
3467 being claimed. If conformance is being claimed to more than one PP, the set of security
3468 objectives for the operational environment must be identical to the union of the security
3469 objectives for the operational environment in the PPs to which conformance is being
3470 claimed with the possible exception as follows:

3471 - a security objective for the operational environment (or part of such security
3472 objective) from one PP can be replaced by the same (part of the) security objective
3473 for the TOE from another PP.

3474 If strict conformance is required by the PP to which conformance is being claimed, no conformance
3475 claim rationale is required. Instead, the evaluator determines whether:

3476 — The ST contains all security objectives for the TOE of the PP to which conformance is being
3477 claimed. Note that it is allowed for the ST under evaluation to have additional security
3478 objectives for the TOE;

3479 — The security objectives for the operational environment in the ST are identical to the security
3480 objectives for the operational environment in the PP to which conformance is being claimed,
3481 with two possible exceptions described in the following two bullet points;

3482 — a security objective for the operational environment (or part of such security objective) from
3483 the PP can be replaced by the same (part of the) security objective stated for the TOE;

3484 — a security objective for the operational environment can be added to the objectives defined in
3485 the PP, if a justification is given, why the new objective neither mitigates a threat (or a part of a
3486 threat) meant to be addressed by security objectives for the TOE in the PP, nor fulfils an OSP
3487 (or part of an OSP) meant to be addressed by security objectives for the TOE in the PP.

- 3488 When examining an ST claiming a PP, which omits security objectives for the operational
 3489 environment from the PP or adds new security objectives for the operational environment, the
 3490 evaluator shall carefully determine, if the conditions given above are fulfilled. The examples given
 3491 for the case of assumptions in the preceding work unit are also valid here.
- 3492 If demonstrable conformance is required by the PP to which conformance is being claimed, the
 3493 evaluator examines the conformance claim rationale to determine that it demonstrates that the
 3494 statement of security objectives of the ST is equivalent or more restrictive than the statement of
 3495 security objectives in the PP to which conformance is being claimed.
- 3496 For this the conformance claim rationale needs to demonstrate that the security objectives in the
 3497 ST are equivalent (or more restrictive) than the security objectives in the PP. This means that:
- 3498 — all TOEs that would meet the security objectives for the TOE in the ST also meet the security
 3499 objectives for the TOE in the PP;
 - 3500 — all operational environments that would meet the security objectives for the operational
 3501 environment in the PP would also meet the security objectives for the operational
 3502 environment in the ST (with one exception in the next bullet);
 - 3503 — besides a set of security objectives for the operational environment in the ST, which are used
 3504 to demonstrate conformance to the set of security objectives defined in the PP, an ST may
 3505 specify further security objectives for the operational environment, but only if these security
 3506 objectives neither affect the original set of security objectives for the TOE nor the security
 3507 objectives for the operational environment as defined in the PP to which conformance is
 3508 claimed."
- 3509 For a composed TOE, the evaluator will consider whether the security objectives of the composed
 3510 TOE are consistent with that specified in the STs for the component TOEs. This is determined in
 3511 terms of demonstrable conformance. In particular, the evaluator examines the conformance claim
 3512 rationale to determine that:
- 3513 a) The statement of security objectives in the dependent TOE ST relevant to any IT in the
 3514 operational environment are consistent with the statement of security objectives for the
 3515 TOE in the base TOE ST. It is not expected that the statement of security objectives for the
 3516 environment within in the dependent TOE ST will cover all aspects of the statement of
 3517 security objectives for the TOE in the base TOE ST.
 - 3518 b) The statement of security objectives in the composed ST is consistent with the statements
 3519 of security objectives in the STs for the component TOEs.
- 3520 If demonstrable conformance is required by the PP, the evaluator examines the conformance claim
 3521 rationale to determine that it demonstrates that the statement of security objectives of the ST is at
 3522 least equivalent to the statement of security objectives in the PP, or component TOE ST in the case
 3523 of a composed TOE ST.
- 3524 ISO/IEC 15408-3 ASE_CCL.1.10C: *The conformance claim rationale shall demonstrate that the*
 3525 *statement of security requirements is consistent with the statement of security requirements in the*
 3526 *PP-Configuration or PPs for which conformance is being claimed.*
- 3527 **10.4.1.3.14 Work unit ASE_CCL.1-12**
- 3528 In this work unit, the term "PP" shall be understood to mean "PP or PP-Configuration component".
- 3529 The evaluator ***shall examine*** the ST to determine that it is consistent, as defined by the
 3530 conformance statement of the PP, with all security requirements in the PPs for which conformance
 3531 is being claimed.

3532 If the ST does not claim conformance to a PP, this work unit is not applicable and therefore
3533 considered to be satisfied.

3534 If the PP to which conformance is being claimed contains functional packages, the evaluator
3535 determines that the SFRs of the ST consist of all SFRs (or hierarchical SFRs) of all mandatory and
3536 all selected optional functional packages.

3537 If packages are used, the rules defined in the following paragraphs concerning exact, strict and
3538 demonstrable conformance also hold for the SFRs taken from the packages.

3539 If exact conformance is required by the PP to which conformance is being claimed, no conformance
3540 claim rationale is required. Instead, the evaluator determines that the statement of security
3541 requirements in the PP to which conformance is being claimed is exactly reproduced in the ST,
3542 with the following allowances:

3543 a) an SFR from the PP may be iterated or refined in the ST,

3544 b) all SFRs that are defined in the PP to which conformance is being claimed as selection-
3545 based upon a particular selection shall be included if and only if that selection on which
3546 inclusion is based is present in the ST. If a selection is not chosen by the ST author, then
3547 the selection-based SFRs associated with that selection are not included in the ST.

3548 c) There are no additional security requirements (SFRs or SARs) that are included in the ST
3549 that are not also present in the PP.

3550 d) In the case where exact conformance is being claimed to multiple PPs, the evaluator
3551 determines there are no additional security requirements included in the ST that are not
3552 in at least one of the PPs, and that all of the requirements (with the allowances described
3553 above) in all of the PPs have been included in the ST.

3554 If strict conformance is required by the PP to which conformance is being claimed, no
3555 conformance claim rationale is required. Instead, the evaluator determines whether the
3556 statement of security requirements in the ST is a superset of or identical to the statement of
3557 security requirements in the PP to which conformance is being claimed (for strict
3558 conformance).

3559 If demonstrable conformance is required by the PP to which conformance is being claimed, the
3560 evaluator examines the conformance claim rationale to determine that it demonstrates that the
3561 statement of security requirements of the ST is equivalent or more restrictive than the statement of
3562 security requirements in the PP to which conformance is being claimed.

3563 For:

3564 — SFRs: The conformance rationale in the ST shall demonstrate that the overall set of
3565 requirements defined by the SFRs in the ST is equivalent (or more restrictive) than the overall
3566 set of requirements defined by the SFRs in the PP. This means that all TOEs that would meet
3567 the requirements defined by the set of all SFRs in the ST would also meet the requirements
3568 defined by the set of all SFRs in the PP;

3569 — SARs: The ST shall contain all SARs in the PP, but may claim additional SARs or replace SARs by
3570 hierarchically stronger SARs. The completion of operations in the ST must be consistent with
3571 that in the PP; either the same completion will be used in the ST as that in the PP or a
3572 completion that makes the SAR more restrictive (the rules of refinement apply).

3573 For a composed TOE, the evaluator will consider whether the security requirements of the
3574 composed TOE are consistent with that specified in the STs for the component TOEs. This is
3575 determined in terms of demonstrable conformance. In particular, the evaluator examines the
3576 conformance rationale to determine that:

3577 a) The statement of security requirements in the dependent TOE ST relevant to any IT in the
 3578 operational environment is consistent with the statement of security requirements for
 3579 the TOE in the base TOE ST. It is not expected that the statement of security requirements
 3580 for the environment within in the dependent TOE ST will cover all aspects of the
 3581 statement of security requirements for the TOE in the base TOE ST, as some SFRs may
 3582 need to be added to the statement of security requirements in the composed TOE ST.
 3583 However, the statement of security requirements in the base should support the
 3584 operation of the dependent component.

3585 b) The statement of security objectives in the dependent TOE ST relevant to any IT in the
 3586 operational environment is consistent with the statement of security requirements for
 3587 the TOE in the base TOE ST. It is not expected that the statement of security objectives for
 3588 the environment within in the dependent TOE ST will cover all aspects of the statement
 3589 of security requirements for the TOE in the base TOE ST.

3590 c) The statement of security requirements in the composed is consistent with the
 3591 statements of security requirements in the STs for the component TOEs.

3592 If demonstrable conformance is required by the PP to which conformance is being claimed, the
 3593 evaluator examines the conformance claim rationale to determine that it demonstrates that the
 3594 statement of security requirements of the ST is at least equivalent to the statement of security
 3595 requirements in the PP, or component TOE ST in the case of a composed TOE ST.

3596 **10.5 Security problem definition (ASE_SPD)**

3597 **10.5.1 Evaluation of sub-activity (ASE_SPD.1)**

3598 **10.5.1.1 Objectives**

3599 The objective of this sub-activity is to determine that the security problem intended to be
 3600 addressed by the TOE and its operational environment is clearly defined.

3601 **10.5.1.2 Input**

3602 The evaluation evidence for this sub-activity is:

3603 a) the ST.

3604 **10.5.1.3 Action ASE_SPD.1.1E**

3605 ISO/IEC 15408-3 ASE_SPD.1.1C: *The security problem definition shall describe the threats.*

3606 **10.5.1.3.1 Work unit ASE_SPD.1-1**

3607 The evaluator **shall check** that the security problem definition describes the threats.

3608 If all security objectives are derived from assumptions and/or OSPs only, the statement of threats
 3609 need not be present in the ST. In this case, this work unit is not applicable and therefore considered
 3610 to be satisfied.

3611 The evaluator determines that the security problem definition describes the threats that must be
 3612 countered by the TOE and/or operational environment.

3613 ISO/IEC 15408-3 ASE_SPD.1.2C: *All threats shall be described in terms of a threat agent, an asset,*
 3614 *and an adverse action.*

3615 **10.5.1.3.2 Work unit ASE_SPD.1-2**

3616 The evaluator **shall examine** the security problem definition to determine that all threats are
3617 described in terms of a threat agent, an asset, and an adverse action.

3618 If all security objectives are derived from assumptions and/or OSPs only, the statement of threats
3619 need not be present in the ST. In this case, this work unit is not applicable and therefore considered
3620 to be satisfied.

3621 Threat agents may be further described by aspects such as expertise, resource, opportunity, and
3622 motivation.

3623 ISO/IEC 15408-3 ASE_SPD.1.3C: *The security problem definition shall describe the OSPs.*

3624 **10.5.1.3.3 Work unit ASE_SPD.1-3**

3625 The evaluator **shall examine** that the security problem definition describes the OSPs.

3626 If all security objectives are derived from assumptions and threats only, OSPs need not be present
3627 in the ST. In this case, this work unit is not applicable and therefore considered to be satisfied.

3628 The evaluator determines that OSP statements are made in terms of rules or guidelines that must
3629 be followed by the TOE and/or its operational environment.

3630 The evaluator determines that each OSP is explained and/or interpreted in sufficient detail to make
3631 it clearly understandable; a clear presentation of policy statements is necessary to permit tracing
3632 security objectives to them.

3633 ISO/IEC 15408-3 ASE_SPD.1.4C: *The security problem definition shall describe the assumptions*
3634 *about the operational environment of the TOE.*

3635 **10.5.1.3.4 Work unit ASE_SPD.1-4**

3636 The evaluator **shall examine** the security problem definition to determine that it describes the
3637 assumptions about the operational environment of the TOE.

3638 If there are no assumptions, this work unit is not applicable and is therefore considered to be
3639 satisfied.

3640 The evaluator determines that each assumption about the operational environment of the TOE is
3641 explained in sufficient detail to enable consumers to determine that their operational environment
3642 matches the assumption. If the assumptions are not clearly understood, the end result may be that
3643 the TOE is used in an operational environment in which it will not function in a secure manner.

3644 **10.6 Security objectives (ASE_OBJ)**

3645 **10.6.1 Evaluation of sub-activity (ASE_OBJ.1)**

3646 **10.6.1.1 Objectives**

3647 The objective of this sub-activity is to determine whether the security objectives for the
3648 operational environment are clearly defined.

3649 **10.6.1.2 Input**

3650 The evaluation evidence for this sub-activity is:

3651 a) the ST.

3652 **10.6.1.3 Action ASE_OBJ.1.1E**

3653 ISO/IEC 15408-3 ASE_OBJ.1.1C: *The statement of security objectives shall describe the security*
 3654 *objectives for the operational environment.*

3655 **10.6.1.3.1 Work unit ASE_OBJ.1-1**

3656 The evaluator ***shall check*** that the statement of security objectives defines the security objectives
 3657 for the operational environment.

3658 The evaluator checks that the security objectives for the operational environment are identified.

3659 **10.6.2 Evaluation of sub-activity (ASE_OBJ.2)**3660 **10.6.2.1 Objectives**

3661 The objective of this sub-activity is to determine whether the security objectives adequately and
 3662 completely address the security problem definition and that the division of this problem between
 3663 the TOE and its operational environment is clearly defined.

3664 **10.6.2.2 Input**

3665 The evaluation evidence for this sub-activity is:

3666 a) the ST.

3667 **10.6.2.3 Action ASE_OBJ.2.1E**

3668 ISO/IEC 15408-3 ASE_OBJ.2.1C: *The statement of security objectives shall describe the security*
 3669 *objectives for the TOE and the security objectives for the operational environment.*

3670 **10.6.2.3.1 Work unit ASE_OBJ.2-1**

3671 The evaluator ***shall check*** that the statement of security objectives defines the security objectives
 3672 for the TOE and the security objectives for the operational environment.

3673 The evaluator checks that both categories of security objectives are clearly identified and
 3674 separated from the other category.

3675 ISO/IEC 15408-3 ASE_OBJ.2.2C: *The security objectives rationale shall trace each security objective*
 3676 *for the TOE back to threats countered by that security objective and OSPs enforced by that security*
 3677 *objective.*

3678 **10.6.2.3.2 Work unit ASE_OBJ.2-2**

3679 The evaluator ***shall check*** that the security objectives rationale traces all security objectives for the
 3680 TOE back to threats countered by the objectives and/or OSPs enforced by the objectives.

3681 Each security objective for the TOE may trace back to threats or OSPs, or a combination of threats
 3682 and OSPs, but it must trace back to at least one threat or OSP.

3683 Failure to trace implies that either the security objectives rationale is incomplete, the security
 3684 problem definition is incomplete, or the security objective for the TOE has no useful purpose.

3685 ISO/IEC 15408-3 ASE_OBJ.2.3C: *The security objectives rationale shall trace each security objective*
 3686 *for the operational environment back to threats countered by that security objective, OSPs enforced*
 3687 *by that security objective, and assumptions upheld by that security objective.*

3688 **10.6.2.3.3 Work unit ASE_OBJ.2-3**

3689 The evaluator **shall check** that the security objectives rationale traces the security objectives for
3690 the operational environment back to threats countered by that security objective, to OSPs enforced
3691 by that security objective, and to assumptions upheld by that security objective.

3692 Each security objective for the operational environment may trace back to threats, OSPs,
3693 assumptions, or a combination of threats, OSPs and/or assumptions, but it must trace back to at
3694 least one threat, OSP or assumption.

3695 Failure to trace implies that either the security objectives rationale is incomplete, the security
3696 problem definition is incomplete, or the security objective for the operational environment has no
3697 useful purpose.

3698 ISO/IEC 15408-3 ASE_OBJ.2.4C: *The security objectives rationale shall demonstrate that the security*
3699 *objectives counter all threats.*

3700 **10.6.2.3.4 Work unit ASE_OBJ.2-4**

3701 The evaluator **shall examine** the security objectives rationale to determine that it justifies for each
3702 threat that the security objectives are suitable to counter that threat.

3703 If no security objectives trace back to the threat, the evaluator action related to this work unit is
3704 assigned a fail verdict.

3705 The evaluator determines that the justification for a threat shows whether the threat is removed,
3706 diminished or mitigated.

3707 The evaluator determines that the justification for a threat demonstrates that the security
3708 objectives are sufficient: if all security objectives that trace back to the threat are achieved, the
3709 threat is removed, sufficiently diminished, or the effects of the threat are sufficiently mitigated.

3710 Note that the tracings from security objectives to threats provided in the security objectives
3711 rationale may be part of a justification, but do not constitute a justification by themselves. Even in
3712 the case that a security objective is merely a statement reflecting the intent to prevent a particular
3713 threat from being realised, a justification is required, but this justification may be as minimal as
3714 "Security Objective X directly counters Threat Y".

3715 The evaluator also determines that each security objective that traces back to a threat is necessary:
3716 when the security objective is achieved it actually contributes to the removal, diminishing or
3717 mitigation of that threat.

3718 ISO/IEC 15408-3 ASE_OBJ.2.5C: *The security objectives rationale shall demonstrate that the security*
3719 *objectives enforce all OSPs.*

3720 **10.6.2.3.5 Work unit ASE_OBJ.2-5**

3721 The evaluator **shall examine** the security objectives rationale to determine that for each OSP it
3722 justifies that the security objectives are suitable to enforce that OSP.

3723 If no security objectives trace back to the OSP, the evaluator action related to this work unit is
3724 assigned a fail verdict.

3725 The evaluator determines that the justification for an OSP demonstrates that the security
3726 objectives are sufficient: if all security objectives that trace back to that OSP are achieved, the OSP
3727 is enforced.

3728 The evaluator also determines that each security objective that traces back to an OSP is necessary:
3729 when the security objective is achieved it actually contributes to the enforcement of the OSP.

3730 Note that the tracings from security objectives to OSPs provided in the security objectives rationale
3731 may be part of a justification, but do not constitute a justification by themselves. In the case that a
3732 security objective is merely a statement reflecting the intent to enforce a particular OSP, a
3733 justification is required, but this justification may be as minimal as "Security Objective X directly
3734 enforces OSP Y".

3735 ISO/IEC 15408-3 ASE_OBJ.2.6C: *The security objectives rationale shall demonstrate that the security*
3736 *objectives for the operational environment uphold all assumptions.*

3737 **10.6.2.3.6 Work unit ASE_OBJ.2-6**

3738 The evaluator ***shall examine*** the security objectives rationale to determine that for each
3739 assumption for the operational environment it contains an appropriate justification that the
3740 security objectives for the operational environment are suitable to uphold that assumption.

3741 If no security objectives for the operational environment trace back to the assumption, the
3742 evaluator action related to this work unit is assigned a fail verdict.

3743 The evaluator determines that the justification for an assumption about the operational
3744 environment of the TOE demonstrates that the security objectives are sufficient: if all security
3745 objectives for the operational environment that trace back to that assumption are achieved, the
3746 operational environment upholds the assumption.

3747 The evaluator also determines that each security objective for the operational environment that
3748 traces back to an assumption about the operational environment of the TOE is necessary: when the
3749 security objective is achieved it actually contributes to the operational environment upholding the
3750 assumption.

3751 Note that the tracings from security objectives for the operational environment to assumptions
3752 provided in the security objectives rationale may be a part of a justification, but do not constitute a
3753 justification by themselves. Even in the case that a security objective of the operational
3754 environment is merely a restatement of an assumption, a justification is required, but this
3755 justification may be as minimal as "Security Objective X directly upholds Assumption Y".

3756 **10.7 Extended components definition (ASE_ECD)**

3757 **10.7.1 Evaluation of sub-activity (ASE_ECD.1)**

3758 **10.7.1.1 Objectives**

3759 The objective of this sub-activity is to determine whether extended components have been clearly
3760 and unambiguously defined, and whether they are necessary, i.e. they may not be clearly expressed
3761 using existing ISO/IEC 15408-2 or ISO/IEC 15408-3 components.

3762 **10.7.1.2 Input**

3763 The evaluation evidence for this sub-activity is:

3764 a) the ST.

3765 **10.7.1.3 Action ASE_ECD.1.1E**

3766 ISO/IEC 15408-3 ASE_ECD.1.1C: *The statement of security requirements shall identify all extended*
3767 *security requirements.*

3768 **10.7.1.3.1 Work unit ASE_ECD.1-1**

3769 The evaluator **shall check** that all security requirements in the statement of security requirements
3770 that are not identified as extended requirements are present in ISO/IEC 15408-2 or in ISO/IEC
3771 15408-3.

3772 ISO/IEC 15408-3 ASE_ECD.1.2C: *The extended components definition shall define an extended*
3773 *component for each extended security requirement.*

3774 **10.7.1.3.2 Work unit ASE_ECD.1-2**

3775 The evaluator **shall check** that the extended components definition defines an extended
3776 component for each extended security requirement.

3777 If the ST does not contain extended security requirements, this work unit is not applicable and
3778 therefore considered to be satisfied.

3779 A single extended component may be used to define multiple iterations of an extended security
3780 requirement, it is not necessary to repeat this definition for each iteration.

3781 ISO/IEC 15408-3 ASE_ECD.1.3C: *The extended components definition shall describe how each*
3782 *extended component is related to the existing ISO/IEC 15408 components, families, and classes.*

3783 **10.7.1.3.3 Work unit ASE_ECD.1-3**

3784 The evaluator **shall examine** the extended components definition to determine that it describes
3785 how each extended component fits into the existing ISO/IEC 15408 components, families, and
3786 classes.

3787 If the ST does not contain extended security requirements, this work unit is not applicable and
3788 therefore considered to be satisfied.

3789 The evaluator determines that each extended component is either:

3790 a) a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 family, or

3791 b) a member of a new family defined in the ST.

3792 If the extended component is a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 family,
3793 the evaluator determines that the extended components definition adequately describes why the
3794 extended component should be a member of that family and how it relates to other components of
3795 that family.

3796 If the extended component is a member of a new family defined in the ST, the evaluator confirms
3797 that the extended component is not appropriate for an existing family.

3798 If the ST defines new families, the evaluator determines that each new family is either:

3799 a) a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 class, or

3800 b) a member of a new class defined in the ST.

3801 If the family is a member of an existing ISO/IEC 15408-2 or ISO/IEC 15408-3 class, the evaluator
3802 determines that the extended components definition adequately describes why the family should
3803 be a member of that class and how it relates to other families in that class.

3804 If the family is a member of a new class defined in the ST, the evaluator confirms that the family is
3805 not appropriate for an existing class.

3806 **10.7.1.3.4 Work unit ASE_ECD.1-4**

3807 The evaluator **shall examine** the extended components definition to determine that each definition
3808 of an extended component identifies all applicable dependencies of that component.

3809 If the ST does not contain extended security requirements, this work unit is not applicable and
3810 therefore considered to be satisfied.

3811 The evaluator confirms that no applicable dependencies have been overlooked by the ST author.

3812 ISO/IEC 15408-3 ASE_ECD.1.4C: *The extended components definition shall use the existing ISO/IEC*
3813 *15408 components, families, classes, and methodology as a model for presentation.*

3814 **10.7.1.3.5 Work unit ASE_ECD.1-5**

3815 The evaluator **shall examine** the extended components definition to determine that each extended
3816 functional component uses the existing ISO/IEC 15408-2 components as a model for presentation.

3817 If the ST does not contain extended SFRs, this work unit is not applicable and therefore considered
3818 to be satisfied.

3819 The evaluator determines that the extended functional component is consistent with ISO/IEC
3820 15408-2 Subclause 6.1.3, **Component structure**.

3821 If the extended functional component uses operations, the evaluator determines that the extended
3822 functional component is consistent with ISO/IEC 15408-1 Subclause 7.1, **Operations**.

3823 If the extended functional component is hierarchical to an existing functional component, the
3824 evaluator determines that the extended functional component is consistent with ISO/IEC 15408-2
3825 Subclause 6.2.1, **Component changes highlighting**.

3826 **10.7.1.3.6 Work unit ASE_ECD.1-6**

3827 The evaluator **shall examine** the extended components definition to determine that each definition
3828 of a new functional family uses the existing ISO/IEC 15408 functional families as a model for
3829 presentation.

3830 If the ST does not define new functional families, this work unit is not applicable and therefore
3831 considered to be satisfied.

3832 The evaluator determines that all new functional families are defined consistent with ISO/IEC
3833 15408-2 Subclause 6.1.2, **Family structure**.

3834 **10.7.1.3.7 Work unit ASE_ECD.1-7**

3835 The evaluator **shall examine** the extended components definition to determine that each definition
3836 of a new functional class uses the existing ISO/IEC 15408 functional classes as a model for
3837 presentation.

3838 If the ST does not define new functional classes, this work unit is not applicable and therefore
3839 considered to be satisfied.

3840 The evaluator determines that all new functional classes are defined consistent with ISO/IEC
3841 15408-2 Subclause 6.1.1, **Class structure**.

3842 **10.7.1.3.8 Work unit ASE_ECD.1-8**

3843 The evaluator ***shall examine*** the extended components definition to determine that each definition
3844 of an extended assurance component uses the existing ISO/IEC 15408-3 components as a model
3845 for presentation.

3846 If the ST does not contain extended SARs, this work unit is not applicable and therefore considered
3847 to be satisfied.

3848 The evaluator determines that the extended assurance component definition is consistent with
3849 ISO/IEC 15408-3 Subclause **6.1.3, Assurance component structure**.

3850 If the extended assurance component uses operations, the evaluator determines that the extended
3851 assurance component is consistent with ISO/IEC 15408-1 Subclause **7.1, Operations**.

3852 If the extended assurance component is hierarchical to an existing assurance component, the
3853 evaluator determines that the extended assurance component is consistent with ISO/IEC 15408-3
3854 Subclause **6.1.3, Assurance component structure**.

3855 **10.7.1.3.9 Work unit ASE_ECD.1-9**

3856 The evaluator ***shall examine*** the extended components definition to determine that, for each
3857 defined extended assurance component, applicable methodology has been provided.

3858 If the ST does not contain extended SARs, this work unit is not applicable and therefore considered
3859 to be satisfied.

3860 The evaluator determines that, for each evaluator action element of each extended SAR, one or
3861 more work units are provided and that successfully performing all work units for a given evaluator
3862 action element will demonstrate that the element has been achieved.

3863 **10.7.1.3.10 Work unit ASE_ECD.1-10**

3864 The evaluator ***shall examine*** the extended components definition to determine that each definition
3865 of a new assurance family uses the existing ISO/IEC 15408 assurance families as a model for
3866 presentation.

3867 If the ST does not define new assurance families, this work unit is not applicable and therefore
3868 considered to be satisfied.

3869 The evaluator determines that all new assurance families are defined consistent with ISO/IEC
3870 15408-3 Subclause **6.1.2, Assurance family structure**.

3871 **10.7.1.3.11 Work unit ASE_ECD.1-11**

3872 The evaluator ***shall examine*** the extended components definition to determine that each definition
3873 of a new assurance class uses the existing ISO/IEC 15408 assurance classes as a model for
3874 presentation.

3875 If the ST does not define new assurance classes, this work unit is not applicable and therefore
3876 considered to be satisfied.

3877 The evaluator determines that all new assurance classes are defined consistent with ISO/IEC
3878 15408-3 Subclause **6.1.1, Assurance class structure**.

3879 ISO/IEC 15408-3 ASE_ECD.1.5C: *The extended components shall consist of measurable and objective*
3880 *elements such that conformance or nonconformance to these elements can be demonstrated.*

3881 **10.7.1.3.12 Work unit ASE_ECD.1-12**

3882 The evaluator ***shall examine*** the extended components definition to determine that each element
 3883 in each extended component is measurable and states objective evaluation requirements, such that
 3884 conformance or nonconformance can be demonstrated.

3885 If the ST does not contain extended security requirements, this work unit is not applicable and
 3886 therefore considered to be satisfied.

3887 The evaluator determines that elements of extended functional components are stated in such a
 3888 way that they are testable, and traceable through the appropriate TSF representations.

3889 The evaluator also determines that elements of extended assurance components avoid the need for
 3890 subjective evaluator judgement.

3891 The evaluator is reminded that whilst being measurable and objective is appropriate for all
 3892 evaluation criteria, it is acknowledged that no formal method exists to prove such properties.
 3893 Therefore the existing ISO/IEC 15408 functional and assurance components are to be used as a
 3894 model for determining what constitutes conformance with this requirement.

3895 **10.7.1.4 Action ASE_ECD.1.2E**

3896 **10.7.1.4.1 Work unit ASE_ECD.1-13**

3897 The evaluator ***shall examine*** the extended components definition to determine that each extended
 3898 component can not be clearly expressed using existing components.

3899 If the ST does not contain extended security requirements, this work unit is not applicable and
 3900 therefore considered to be satisfied.

3901 The evaluator should take components from ISO/IEC 15408-2 and ISO/IEC 15408-3, other
 3902 extended components that have been defined in the ST, combinations of these components, and
 3903 possible operations on these components into account when making this determination.

3904 The evaluator is reminded that the role of this work unit is to preclude unnecessary duplication of
 3905 components, that is, components that may be clearly expressed by using other components. The
 3906 evaluator should not undertake an exhaustive search of all possible combinations of components
 3907 including operations in an attempt to find a way to express the extended component by using
 3908 existing components.

3909 **10.8 Security requirements (ASE_REQ)**

3910 **10.8.1 Evaluation of sub-activity (ASE_REQ.1)**

3911 **10.8.1.1 Objectives**

3912 The objective of this sub-activity is to determine whether the SFRs and SARs are clear,
 3913 unambiguous and well-defined, whether they are internally consistent, and whether the SFRs
 3914 counter the threats and implement the organisational security policies of the TOE..

3915 **10.8.1.2 Input**

3916 The evaluation evidence for this sub-activity is:

3917 a) the ST.

3918 **10.8.1.3 Action ASE_REQ.1.1E**

3919 ISO/IEC 15408-3 ASE_REQ.1.1C: *The statement of security requirements shall describe the SFRs and*
3920 *the SARs.*

3921 **10.8.1.3.1 Work unit ASE_REQ.1-1**

3922 The evaluator **shall check** that the statement of security requirements describes the SFRs.

3923 The evaluator determines that each SFR is identified by one of the following means:

- 3924 a) by reference to an individual component in ISO/IEC 15408-2;
- 3925 b) by reference to an extended component in the extended components definition of the ST;
- 3926 c) by reference to a PP that the ST claims to be conformant with;
- 3927 d) by reference to a security requirements package that the ST claims to be conformant with;
- 3928 e) by reproduction in the ST.

3929 It is not required to use the same means of identification for all SFRs.

3930 **10.8.1.3.2 Work unit ASE_REQ.1-2**

3931 The evaluator **shall check** that the statement of security requirements describes the SARs.

3932 The evaluator determines that each SAR is identified by one of the following means:

- 3933 a) by reference to an individual component in ISO/IEC 15408-3;
- 3934 b) by reference to an extended component in the extended components definition of the ST;
- 3935 c) by reference to a PP that the ST claims to be conformant with;
- 3936 d) by reference to a security requirements package that the ST claims to be conformant with;
- 3937 e) by reproduction in the ST.

3938 It is not required to use the same means of identification for all SARs.

3939 ISO/IEC 15408-3 ASE_REQ.1.2C: *All subjects, objects, operations, security attributes, external entities*
3940 *and other terms that are used in the SFRs and the SARs shall be defined.*

3941 **10.8.1.3.3 Work unit ASE_REQ.1-3**

3942 The evaluator **shall examine** the ST to determine that all subjects, objects, operations, security
3943 attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

3944 The evaluator determines that the ST defines all:

- 3945 — (types of) subjects and objects that are used in the SFRs;
- 3946 — (types of) security attributes of subjects, users, objects, information, sessions and/or resources,
3947 possible values that these attributes may take and any relations between these values (e.g.
3948 top_secret is “higher” than secret);

- 3949 — (types of) operations that are used in the SFRs, including the effects of these operations;
- 3950 — (types of) external entities in the SFRs;
- 3951 — other terms that are introduced in the SFRs and/or SARs by completing operations, if these
3952 terms are not immediately clear, or are used outside their dictionary definition.
- 3953 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no
3954 misunderstanding may occur due to the introduction of vague terms. This work unit should not be
3955 taken into extremes, by forcing the ST writer to define every single word. The general audience of a
3956 set of security requirements should be assumed to have a reasonable knowledge of IT, security and
3957 "Evaluation criteria for IT security".
- 3958 All of the above may be presented in groups, classes, roles, types or other groupings or
3959 characterisations that allow easy understanding.
- 3960 The evaluator is reminded that these lists and definitions do not have to be part of the statement of
3961 security requirements, but may be placed (in part or in whole) in different subclauses. This may be
3962 especially applicable if the same terms are used in the rest of the ST.
- 3963 ISO/IEC 15408-3 ASE_REQ.1.3C: *The statement of security requirements shall include a natural*
3964 *language description, part of which describes how the SFRs combine together to provide security*
3965 *functionality in terms of the architecture that is visible to Administrators and other users.*
- 3966 **10.8.1.3.4 Work unit ASE_REQ.1-4**
- 3967 The evaluator **shall check** that the statement of security requirements includes a natural language
3968 description, part of which describes how the SFRs combine together to provide security
3969 functionality in terms of the architecture that is visible to Administrators and other users.
- 3970 The description is intended to make clear connections between SFRs and to provide a view of how
3971 they provide security functionality that is recognizable to Administrators and other types of
3972 user. The description in terms of the architecture that is "visible to Administrators and other
3973 users" means that the description must relate the security behavior to visible elements, but
3974 the mechanisms themselves need not be visible. For example: when describing authentication
3975 using a biometric mechanism, the calculation of the match or score might not be visible, but
3976 (a) might relate to a referenced description of a matching algorithm, (b) might be based on
3977 specific template files maintained by the Administrator, and (c) will result in acceptance or
3978 rejection of the authentication attempt – therefore the description might make use of any or
3979 all of these items (a) – (c). No specific format for this information is prescribed, and the
3980 description need not all be located alongside the SFRs themselves (e.g. some of it might be in
3981 the ST Introduction and/or in the TSS). The intention of the requirement is to make the
3982 meaning of the SFRs clearer and more easily understood by readers of the ST who may not
3983 have deep knowledge of the CC but who are familiar with the product type.
- 3984 The evaluator determines that all operations are identified in each SFR or SAR where such an
3985 operation is used. This includes both completed operations and uncompleted operations.
3986 Identification may be achieved by typographical distinctions, or by explicit identification in the
3987 surrounding text, or by any other distinctive means.
- 3988 ISO/IEC 15408-3 ASE_REQ.1.4C: *The statement of security requirements shall identify all operations*
3989 *on the security requirements.*
- 3990 **10.8.1.3.5 Work unit ASE_REQ.1-5**
- 3991 The evaluator **shall check** that the statement of security requirements identifies all operations on
3992 the security requirements.

3993 The evaluator determines that all operations are identified in each SFR or SAR where such an
3994 operation is used. Identification may be achieved by typographical distinctions, or by explicit
3995 identification in the surrounding text, or by any other distinctive means.

3996 ISO/IEC 15408-3 ASE_REQ.1.5C: *All operations shall be performed correctly.*

3997 **10.8.1.3.6 Work unit ASE_REQ.1-6**

3998 The evaluator ***shall examine*** the statement of security requirements to determine that all
3999 assignment operations are performed correctly.

4000 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4001 **Guidance for Operations.**

4002 **10.8.1.3.7 Work unit ASE_REQ.1-7**

4003 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration
4004 operations are performed correctly.

4005 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4006 **Guidance for Operations.**

4007 **10.8.1.3.8 Work unit ASE_REQ.1-8**

4008 The evaluator ***shall examine*** the statement of security requirements to determine that all selection
4009 operations are performed correctly.

4010 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4011 **Guidance for Operations.**

4012 **10.8.1.3.9 Work unit ASE_REQ.1-9**

4013 The evaluator ***shall examine*** the statement of security requirements to determine that all
4014 refinement operations are performed correctly.

4015 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4016 **Guidance for Operations.**

4017 ISO/IEC 15408-3 ASE_REQ.1.6C: *Each dependency of the security requirements shall either be*
4018 *satisfied, or the security requirements rationale shall justify the dependency not being satisfied.*

4019 **10.8.1.3.10 Work unit ASE_REQ.1-10**

4020 The evaluator ***shall examine*** the statement of security requirements to determine that each
4021 dependency of the security requirements is either satisfied, or that a security requirements
4022 rationale is provided which justifies the dependency not being satisfied.

4023 A dependency is satisfied by the inclusion of the relevant component (or one that is hierarchical to
4024 it) within the statement of security requirements. The component used to satisfy the dependency
4025 should, if necessary, be modified by operations to ensure that it actually satisfies that dependency.

4026 A justification that a dependency is not met should address either:

4027 a) why the dependency is not necessary or useful, in which case no further information is
4028 required; or

- 4029 b) that the dependency has been addressed by the operational environment of the TOE, in
4030 which case the justification should describe how the security objectives for the
4031 operational environment address this dependency.
- 4032 ISO/IEC 15408-3 ASE_REQ.1.7C: The security requirements rationale shall trace each SFR back to
4033 the threats countered by that SFR and OSPs enforced by that SFR.
- 4034 **10.8.1.3.11 Work unit ASE_REQ.1-11**
- 4035 The evaluator ***shall check*** that the security requirements rationale traces each SFR back to the
4036 threats countered by that SFR and OSPs enforced by that SFR.
- 4037 The evaluator determines that each SFR is traced back to at least one threat or OSP for the TOE.
- 4038 Failure to trace implies that either the security requirements rationale is incomplete, the security
4039 objectives for the TOE are incomplete, or the SFR has no useful purpose.
- 4040 There is no prescribed location for this part of the rationale: for example, the relevant parts may be
4041 located under each threat and OSP in order to help make the security argument clearer and easier
4042 to read.
- 4043 ISO/IEC 15408-3 ASE_REQ.1.8C: *The security requirements rationale shall trace each security*
4044 *objective for the operational environment back to threats countered by that security objective, OSPs*
4045 *enforced by that security objective, and assumptions upheld by that security objective.*
- 4046 **10.8.1.3.12 Work unit ASE_REQ.1-12**
- 4047 The evaluator ***shall check*** that the security objectives requirements rationale traces the security
4048 objectives for the operational environment back to threats countered by that security objective, to
4049 OSPs enforced by that security objective, and to assumptions upheld by that security objective.
- 4050 Each security objective for the operational environment may trace back to threats, OSPs,
4051 assumptions, or a combination of threats, OSPs and/or assumptions, but it must trace back to at
4052 least one threat, OSP or assumption.
- 4053 Failure to trace implies that either the security objectives requirements rationale is incomplete, the
4054 security problem definition is incomplete, or the security objective for the operational
4055 environment has no useful purpose.
- 4056 There is no prescribed location for this part of the rationale: for example, the relevant parts may be
4057 located under each threat, OSP and assumption in order to help make the security argument
4058 clearer and easier to read.
- 4059 ISO/IEC 15408-3 ASE_REQ.1.9C: The security requirements rationale shall demonstrate that the
4060 SFRs (in conjunction with the security objectives for the environment) counter all threats for the
4061 TOE.
- 4062 **10.8.1.3.13 Work unit ASE_REQ.1-13**
- 4063 The evaluator ***shall examine*** the security requirements rationale to determine that for each threat
4064 it demonstrates that the SFRs are suitable to meet that threat.
- 4065 If no SFRs trace back to a threat, the evaluator action related to this work unit is assigned a fail
4066 verdict.
- 4067 The evaluator determines that the justification for a threat shows whether the threat is removed,
4068 diminished or mitigated.

4069 The evaluator determines that the justification for a threat demonstrates that the SFRs are
 4070 sufficient: if all SFRs that trace back to the threat are achieved then, in the context of any applicable
 4071 OSPs and assumptions, the threat is removed, sufficiently diminished, or the effects of the threat
 4072 are sufficiently mitigated.

4073 Note that simply listing in the security requirements rationale the SFRs associated with each threat
 4074 may be part of a justification, but does not constitute a justification by itself. A descriptive
 4075 justification is required, although in simple cases this justification may be as minimal as "SFR X
 4076 directly counters Threat Y".

4077 The evaluator also determines that each SFR that traces back to a threat is necessary: when the SFR
 4078 is implemented it actually contributes to the removal, diminishing or mitigation of that threat.

4079 ISO/IEC 15408-3 ASE_REQ.1.10C: The security requirements rationale shall demonstrate that the
 4080 SFRs (in conjunction with the security objectives for the environment) enforce all OSPs for the TOE.

4081 **10.8.1.3.14 Work unit ASE_REQ.1-14**

4082 The evaluator ***shall examine*** the security requirements rationale to determine that for each OSP it
 4083 justifies that the SFRs are suitable to enforce that OSP.

4084 If no SFRs or security objectives for the operational environment trace back to the OSP, the
 4085 evaluator action related to this work unit is assigned a fail verdict.

4086 The evaluator determines that the justification for an OSP demonstrates that the security
 4087 objectives are sufficient: if all SFRs that trace back to that OSP are achieved then, in the context of
 4088 any applicable assumptions, the OSP is enforced.

4089 The evaluator also determines that each SFR that traces back to an OSP is necessary: when the SFR
 4090 is implemented it actually contributes to the enforcement of the OSP.

4091 Note that simply listing in the security requirements rationale the SFRs associated with each OSP
 4092 may be part of a justification, but does not constitute a justification by itself. A descriptive
 4093 justification is required, although in simple cases this justification may be as minimal as "SFR X
 4094 directly enforces OSP Y".

4095 ISO/IEC 15408-3 ASE_REQ.1.11C: The security requirements rationale shall demonstrate that the
 4096 SFRs (in conjunction with the security objectives for the environment) enforce all OSPs for the TOE.

4097 **10.8.1.3.15 Work unit ASE_REQ.1-15**

4098 The evaluator ***shall examine*** the security requirements rationale to determine that for each
 4099 assumption for the operational environment it contains an appropriate justification that the
 4100 security objectives for the operational environment are suitable to uphold that assumption.

4101 If no security objectives for the operational environment trace back to the assumption, the
 4102 evaluator action related to this work unit is assigned a fail verdict.

4103 The evaluator determines that the justification for an assumption about the operational
 4104 environment of the TOE demonstrates that the security objectives are sufficient: if all security
 4105 objectives for the operational environment that trace back to that assumption are achieved, the
 4106 operational environment upholds the assumption.

4107 The evaluator also determines that each security objective for the operational environment that
 4108 traces back to an assumption about the operational environment of the TOE is necessary: when the
 4109 security objective is achieved it actually contributes to the operational environment upholding the
 4110 assumption.

4111 Note that simply listing in the security requirements rationale the security objectives for the
 4112 operational environment associated with each assumption may be a part of a justification, but does
 4113 not constitute a justification by itself. A descriptive justification is required, although in simple
 4114 cases this justification may be as minimal as “Security Objective X directly upholds Assumption Y”.

4115 ISO/IEC 15408-3 ASE_REQ.1.12C: *The statement of security requirements shall be internally*
 4116 *consistent.*

4117 **10.8.1.3.16 Work unit ASE_REQ.1-16**

4118 The evaluator ***shall examine*** the statement of security requirements to determine that it is
 4119 internally consistent.

4120 The evaluator determines that the combined set of all SFRs and SARs is internally consistent.

4121 The evaluator determines that on all occasions where different security requirements apply to the
 4122 same types of developer evidence, events, operations, data, tests to be performed etc. or to “all
 4123 objects”, “all subjects” etc., that these requirements do not conflict.

4124 Some possible conflicts are:

4125 a) an extended SAR specifying that the design of a certain cryptographic algorithm is to be
 4126 kept secret, and another extended SAR specifying an open source review;

4127 b) **FAU_GEN.1 Audit data generation** specifying that subject identity is to be logged,
 4128 **FDP_ACC.1 Subset access control** specifying who has access to these logs, and **FPR_UNO.1**
 4129 **Unobservability** specifying that some actions of subjects should be unobservable to other
 4130 subjects. If the subject that should not be able to see an activity may access logs of this
 4131 activity, these SFRs conflict;

4132 c) **FDP_RIP.1 Subset residual information protection** specifying deletion of information no
 4133 longer needed, and **FDP_ROL.1 Basic rollback** specifying that a TOE may return to a
 4134 previous state. If the information that is needed for the rollback to the previous state has
 4135 been deleted, these requirements conflict;

4136 d) Multiple iterations of **FDP_ACC.1 Subset access control** especially where some iterations
 4137 cover the same subjects, objects, or operations. If one access control SFR allows a subject
 4138 to perform an operation on an object, while another access control SFR does not allow
 4139 this, these requirements conflict.

4140 **10.8.2 Evaluation of sub-activity (ASE_REQ.2)**

4141 **10.8.2.1 Objectives**

4142 The objective of this sub-activity is to determine whether the SFRs and SARs are clear,
 4143 unambiguous and well-defined, whether they are internally consistent, and whether the SFRs meet
 4144 the security objectives of the TOE.

4145 **10.8.2.2 Input**

4146 The evaluation evidence for this sub-activity is:

4147 a) the ST.

4148 **10.8.2.3 Action ASE_REQ.2.1E**

4149 ISO/IEC 15408-3 ASE_REQ.2.1C: *The statement of security requirements shall describe the SFRs and*
 4150 *the SARs.*

4151 **10.8.2.3.1 Work unit ASE_REQ.2-1**

4152 The evaluator **shall check** that the statement of security requirements describes the SFRs.

4153 The evaluator determines that each SFRs is identified by one of the following means:

- 4154 a) by reference to an individual component in ISO/IEC 15408-2;
- 4155 b) by reference to an extended component in the extended components definition of the ST;
- 4156 c) by reference to an individual component in a PP that the ST claims to be conformant with;
- 4157 d) by reference to an individual component in a security requirements package that the ST
- 4158 claims to be conformant with;
- 4159 e) by reproduction in the ST.

4160 It is not required to use the same means of identification for all SFRs.

4161 **10.8.2.3.2 Work unit ASE_REQ.2-2**

4162 The evaluator **shall check** that the statement of security requirements describes the SARs.

4163 The evaluator determines that all SARs are identified by one of the following means:

- 4164 a) by reference to an individual component in ISO/IEC 15408-3;
- 4165 b) by reference to an extended component in the extended components definition of the ST;
- 4166 c) by reference to an individual component in a PP that the ST claims to be conformant with;
- 4167 d) by reference to an individual component in a security requirements package that the ST
- 4168 claims to be conformant with;
- 4169 e) by reproduction in the ST.

4170 It is not required to use the same means of identification for all SARs.

4171 ISO/IEC 15408-3 ASE_REQ.2.2C: *All subjects, objects, operations, security attributes, external entities*
4172 *and other terms that are used in the SFRs and the SARs shall be defined.*

4173 **10.8.2.3.3 Work unit ASE_REQ.2-3**

4174 The evaluator **shall examine** the ST to determine that all subjects, objects, operations, security
4175 attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

4176 The evaluator determines that the ST defines all:

- 4177 — (types of) subjects and objects that are used in the SFRs;
- 4178 — (types of) security attributes of subjects, users, objects, information, sessions and/or resources,
4179 possible values that these attributes may take and any relations between these values (e.g.
4180 top_secret is “higher” than secret);
- 4181 — (types of) operations that are used in the SFRs, including the effects of these operations;
- 4182 — (types of) external entities in the SFRs;

- 4183 — other terms that are introduced in the SFRs and/or SARs by completing operations, if these
4184 terms are not immediately clear, or are used outside their dictionary definition.
- 4185 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no
4186 misunderstanding may occur due to the introduction of vague terms. This work unit should not be
4187 taken into extremes, by forcing the ST writer to define every single word. The general audience of a
4188 set of security requirements should be assumed to have a reasonable knowledge of IT, security and
4189 "Evaluation criteria for IT security".
- 4190 All of the above may be presented in groups, classes, roles, types or other groupings or
4191 characterisations that allow easy understanding.
- 4192 The evaluator is reminded that these lists and definitions do not have to be part of the statement of
4193 security requirements, but may be placed (in part or in whole) in different subclauses. This may be
4194 especially applicable if the same terms are used in the rest of the ST.
- 4195 ISO/IEC 15408-3 ASE_REQ.2.3C: *The statement of security requirements shall identify all operations*
4196 *on the security requirements.*
- 4197 **10.8.2.3.4 Work unit ASE_REQ.2-4**
- 4198 The evaluator ***shall check*** that the statement of security requirements identifies all operations on
4199 the security requirements.
- 4200 The evaluator determines that all operations are identified in each SFR or SAR where such an
4201 operation is used. Identification may be achieved by typographical distinctions, or by explicit
4202 identification in the surrounding text, or by any other distinctive means.
- 4203 ISO/IEC 15408-3 ASE_REQ.2.4C: *All operations shall be performed correctly.*
- 4204 **10.8.2.3.5 Work unit ASE_REQ.2-5**
- 4205 The evaluator ***shall examine*** the statement of security requirements to determine that all
4206 assignment operations are performed correctly.
- 4207 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4208 **Guidance for Operations.**
- 4209 **10.8.2.3.6 Work unit ASE_REQ.2-6**
- 4210 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration
4211 operations are performed correctly.
- 4212 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4213 **Guidance for Operations.**
- 4214 **10.8.2.3.7 Work unit ASE_REQ.2-7**
- 4215 The evaluator ***shall examine*** the statement of security requirements to determine that all selection
4216 operations are performed correctly.
- 4217 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4218 **Guidance for Operations.**
- 4219 **10.8.2.3.8 Work unit ASE_REQ.2-8**
- 4220 The evaluator ***shall examine*** the statement of security requirements to determine that all
4221 refinement operations are performed correctly.

4222 Guidance on the correct performance of operations may be found in ISO/IEC 15408-1 Annex C,
4223 **Guidance for Operations.**

4224 ISO/IEC 15408-3 ASE_REQ.2.5C: *Each dependency of the security requirements shall either be*
4225 *satisfied, or the security requirements rationale shall justify the dependency not being satisfied.*

4226 **10.8.2.3.9 Work unit ASE_REQ.2-9**

4227 The evaluator **shall examine** the statement of security requirements to determine that each
4228 dependency of the security requirements is either satisfied, or that the security requirements
4229 rationale justifies the dependency not being satisfied.

4230 A dependency is satisfied by the inclusion of the relevant component (or one that is hierarchical to
4231 it) within the statement of security requirements. The component used to satisfy the dependency
4232 should, if necessary, be modified by operations to ensure that it actually satisfies that dependency.

4233 A justification that a dependency is not met should address either:

4234 a) why the dependency is not necessary or useful, in which case no further information is
4235 required; or

4236 b) that the dependency has been addressed by the operational environment of the TOE, in
4237 which case the justification should describe how the security objectives for the
4238 operational environment address this dependency.

4239 ISO/IEC 15408-3 ASE_REQ.2.6C: *The security requirements rationale shall trace each SFR back to the*
4240 *security objectives for the TOE.*

4241 **10.8.2.3.10 Work unit ASE_REQ.2-10**

4242 The evaluator **shall check** that the security requirements rationale traces each SFR back to the
4243 security objectives for the TOE.

4244 The evaluator determines that each SFR is traced back to at least one security objective for the TOE.

4245 Failure to trace implies that either the security requirements rationale is incomplete, the security
4246 objectives for the TOE are incomplete, or the SFR has no useful purpose.

4247 ISO/IEC 15408-3 ASE_REQ.2.7C: *The security requirements rationale shall demonstrate that the*
4248 *SFRs meet all security objectives for the TOE.*

4249 **10.8.2.3.11 Work unit ASE_REQ.2-11**

4250 The evaluator **shall examine** the security requirements rationale to determine that for each
4251 security objective for the TOE it demonstrates that the SFRs are suitable to meet that security
4252 objective for the TOE.

4253 If no SFRs trace back to the security objective for the TOE, the evaluator action related to this work
4254 unit is assigned a fail verdict.

4255 The evaluator determines that the justification for a security objective for the TOE demonstrates
4256 that the SFRs are sufficient: if all SFRs that trace back to the objective are satisfied, the security
4257 objective for the TOE is achieved.

4258 The evaluator also determines that each SFR that traces back to a security objective for the TOE is
4259 necessary: when the SFR is satisfied, it actually contributes to achieving the security objective.

- 4260 Note that the tracings from SFRs to security objectives for the TOE provided in the security
4261 requirements rationale may be a part of the justification, but do not constitute a justification by
4262 themselves.
- 4263 ISO/IEC 15408-3 ASE_REQ.2.8C: *The security requirements rationale shall explain why the SARs*
4264 *were chosen.*
- 4265 **10.8.2.3.12 Work unit ASE_REQ.2-12**
- 4266 The evaluator ***shall check*** that the security requirements rationale explains why the SARs were
4267 chosen.
- 4268 The evaluator is reminded that any explanation is correct, as long as it is coherent and neither the
4269 SARs nor the explanation have obvious inconsistencies with the remainder of the ST.
- 4270 An example of an obvious inconsistency between the SARs and the remainder of the ST would be to
4271 have threat agents that are very capable, but an AVA_VAN SAR that does not protect against these
4272 threat agents.
- 4273 ISO/IEC 15408-3 ASE_REQ.2.9C: *The statement of security requirements shall be internally*
4274 *consistent.*
- 4275 **10.8.2.3.13 Work unit ASE_REQ.2-13**
- 4276 The evaluator ***shall examine*** the statement of security requirements to determine that it is
4277 internally consistent.
- 4278 The evaluator determines that the combined set of all SFRs and SARs is internally consistent.
- 4279 The evaluator determines that on all occasions where different security requirements apply to the
4280 same types of developer evidence, events, operations, data, tests to be performed etc. or to “all
4281 objects”, “all subjects” etc., that these requirements do not conflict.
- 4282 Some possible conflicts are:
- 4283 a) an extended SAR specifying that the design of a certain cryptographic algorithm is to be
4284 kept secret, and another extended assurance requirement specifying an open source
4285 review;
- 4286 b) **FAU_GEN.1 Audit data generation** specifying that subject identity is to be logged,
4287 **FDP_ACC.1 Subset access control** specifying who has access to these logs, and **FPR_UNO.1**
4288 **Unobservability** specifying that some actions of subjects should be unobservable to other
4289 subjects. If the subject that should not be able to see an activity may access logs of this
4290 activity, these SFRs conflict;
- 4291 c) **FDP_RIP.1 Subset residual information protection** specifying deletion of information no
4292 longer needed, and **FDP_ROL.1 Basic rollback** specifying that a TOE may return to a
4293 previous state. If the information that is needed for the rollback to the previous state has
4294 been deleted, these requirements conflict;
- 4295 d) Multiple iterations of **FDP_ACC.1 Subset access control** especially where some iterations
4296 cover the same subjects, objects, or operations. If one access control SFR allows a subject
4297 to perform an operation on an object, while another access control SFR does not allow
4298 this, these requirements conflict.

4299 **10.9 TOE summary specification (ASE_TSS)**

4300 **10.9.1 Evaluation of sub-activity (ASE_TSS.1)**

4301 **10.9.1.1 Objectives**

4302 The objective of this sub-activity is to determine whether the TOE summary specification
4303 addresses all SFRs, and whether the TOE summary specification is consistent with other narrative
4304 descriptions of the TOE.

4305 **10.9.1.2 Input**

4306 The evaluation evidence for this sub-activity is:

4307 a) the ST.

4308 **10.9.1.3 Action ASE_TSS.1.1E**

4309 ISO/IEC 15408-3 ASE_TSS.1.1C: *The TOE summary specification shall describe how the TOE meets*
4310 *each SFR.*

4311 **10.9.1.3.1 Work unit ASE_TSS.1-1**

4312 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how
4313 the TOE meets each SFR.

4314 The evaluator determines that the TOE summary specification provides, for each SFR from the
4315 statement of security requirements, a description on how that SFR is met.

4316 The evaluator is reminded that the objective of each description is to provide potential consumers
4317 of the TOE with a high-level view of how the developer intends to satisfy each SFR and that the
4318 descriptions therefore should not be overly detailed. Often several SFRs will be implemented in
4319 one context; for instance a password authentication mechanism may implement FIA_UAU.1,
4320 FIA_SOS.1 and FIA_UID.1. Therefore usually the TSS will not consist of a long list with texts for each
4321 single SFR, but complete groups of SFRs may be covered by one text passage.

4322 For a composed TOE, the evaluator also determines that it is clear which component provides each
4323 SFR or how the components combine to meet each SFR.

4324 **10.9.1.4 Action ASE_TSS.1.2E**

4325 **10.9.1.4.1 Work unit ASE_TSS.1-2**

4326 The evaluator ***shall examine*** the TOE summary specification to determine that it is consistent with
4327 the TOE overview and the TOE description.

4328 The TOE overview, TOE description, and TOE summary specification describe the TOE in a
4329 narrative form at increasing levels of detail. These descriptions therefore need to be consistent.

4330 **10.9.2 Evaluation of sub-activity (ASE_TSS.2)**

4331 **10.9.2.1 Objectives**

4332 The objective of this sub-activity is to determine whether the TOE summary specification
4333 addresses all SFRs, whether the TOE summary specification addresses interference, logical
4334 tampering and bypass, and whether the TOE summary specification is consistent with other
4335 narrative descriptions of the TOE.

4336 **10.9.2.2 Input**

4337 The evaluation evidence for this sub-activity is:

4338 a) the ST.

4339 **10.9.2.3 Action ASE_TSS.2.1E**4340 ISO/IEC 15408-3 ASE_TSS.2.1C: *The TOE summary specification shall describe how the TOE meets*
4341 *each SFR.*4342 **10.9.2.3.1 Work unit ASE_TSS.2-1**4343 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how
4344 the TOE meets each SFR.4345 The evaluator determines that the TOE summary specification provides, for each SFR from the
4346 statement of security requirements, a description on how that SFR is met.4347 The evaluator is reminded that the objective of each description is to provide potential consumers
4348 of the TOE with a high-level view of how the developer intends to satisfy each SFR and that the
4349 descriptions therefore should not be overly detailed. Often several SFRs will be implemented in
4350 one context; for instance a password authentication mechanism may implement FIA_UAU.1,
4351 FIA_SOS.1 and FIA_UID.1. Therefore usually the TSS will not consist of a long list with texts for each
4352 single SFR, but complete groups of SFRs may be covered by one text passage.4353 For a composed TOE, the evaluator also determines that it is clear which component provides each
4354 SFR or how the components combine to meet each SFR.4355 ISO/IEC 15408-3 ASE_TSS.2.2C: *The TOE summary specification shall describe how the TOE protects*
4356 *itself against interference and logical tampering.*4357 **10.9.2.3.2 Work unit ASE_TSS.2-2**4358 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how
4359 the TOE protects itself against interference and logical tampering.4360 The evaluator is reminded that the objective of each description is to provide potential consumers
4361 of the TOE with a high-level view of how the developer intends to provide protection against
4362 interference and logical tampering and that the descriptions therefore should not be overly
4363 detailed.4364 For a composed TOE, the evaluator also determines that it is clear which component provides the
4365 protection or how the components combine to provide protection.4366 ISO/IEC 15408-3 ASE_TSS.2.3C: *The TOE summary specification shall describe how the TOE protects*
4367 *itself against bypass.*4368 **10.9.2.3.3 Work unit ASE_TSS.2-3**4369 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how
4370 the TOE protects itself against bypass.4371 The evaluator is reminded that the objective of each description is to provide potential consumers
4372 of the TOE with a high-level view of how the developer intends to provide protection against
4373 bypass and that the descriptions therefore should not be overly detailed.

4374 For a composed TOE, the evaluator also determines that it is clear which component provides the
4375 protection or how the components combine to provide protection.

4376 10.9.2.4 Action ASE_TSS.2.2E

4377 10.9.2.4.1 Work unit ASE_TSS.2-4

4378 The evaluator **shall examine** the TOE summary specification to determine that it is consistent with
4379 the TOE overview and the TOE description.

4380 The TOE overview, TOE description, and TOE summary specification describe the TOE in a
4381 narrative form at increasing levels of detail. These descriptions therefore need to be consistent.

4382 10.10 [PLACE-HOLDER] ST Additional Module Analysis (ASE_AMA)

4383 If the modularity approach included in ASE_AMA, ADV_MTC, ATE_MTK, ATE_MTT remains in
4384 ISO/IEC 15408-x then work units will be required to cover these.

4385 **Suggestions for text would be welcomed in response to CD1 review. If none are received then**
4386 **this topic will be left to the next revision.**

4387

4388 11 Class ADV: Development

4389 11.1 Introduction

4390 The purpose of the development activity is to assess the design documentation in terms of its
4391 adequacy to understand how the TSF meets the SFRs and how the implementation of these SFRs
4392 cannot be tampered with or bypassed. This understanding is achieved through examination of
4393 increasingly refined descriptions of the TSF design documentation. Design documentation consists
4394 of a functional specification (which describes the interfaces of the TSF), a TOE design description
4395 (which describes the architecture of the TSF in terms of how it works in order to perform the
4396 functions related to the SFRs being claimed), and an implementation description (a source code
4397 level description). In addition, there is a security architecture description (which describes the
4398 architectural properties of the TSF to explain how its security enforcement cannot be compromised
4399 or bypassed), an internals description (which describes how the TSF was constructed in a manner
4400 that encourages understandability), and a security policy model (which formally describes the
4401 security policies enforced by the TSF).

4402 11.2 Application notes

4403 ISO/IEC 15408 requirements for design documentation are levelled by the amount, and detail of
4404 information provided, and the degree of formality of the presentation of the information. At lower
4405 levels, the most security-critical portions of the TSF are described with the most detail, while less
4406 security-critical portions of the TSF are merely summarised; added assurance is gained by
4407 increasing the amount of information about the most security-critical portions of the TSF, and
4408 increasing the details about the less security-critical portions. The most assurance is achieved
4409 when thorough details and information of all portions are provided.

4410 ISO/IEC 15408 considers a document's degree of formality (that is, whether it is informal or
4411 semiformal) to be hierarchical. An informal document is one that is expressed in a natural language.
4412 The methodology does not dictate the specific language that must be used; that issue is left for the
4413 scheme. The following paragraphs differentiate the contents of the different informal documents.

4414 A functional specification provides a description of the purpose and method-of-use of interfaces to
4415 the TSF. For example, if an operating system presents the user with a means of self-identification,
4416 of creating files, of modifying or deleting files, of setting permissions defining what other users may

4417 access files, and of communicating with remote machines, its functional specification would
 4418 contain descriptions of each of these and how they are realised through interactions with the
 4419 externally-visible interfaces to the TSF. If there is also audit functionality that detects and record
 4420 the occurrences of such events, descriptions of this audit functionality would also be expected to be
 4421 part of the functional specification; while this functionality is technically not directly invoked by
 4422 the user at the external interface, it certainly is affected by what occurs at the user's external
 4423 interface.

4424 A design description is expressed in terms of logical divisions (subsystems or modules) that each
 4425 provide a comprehensible service or function. For example, a firewall might be composed of
 4426 subsystems that deal with packet filtering, with remote administration, with auditing, and with
 4427 connection-level filtering. The design description of the firewall would describe the actions that are
 4428 taken, in terms of what actions each subsystem takes when an incoming packet arrives at the
 4429 firewall.

4430 **11.3 Security Architecture (ADV_ARC)**

4431 **11.3.1 Evaluation of sub-activity (ADV_ARC.1)**

4432 **11.3.1.1 Objectives**

4433 The objective of this sub-activity is to determine whether the TSF is structured such that it cannot
 4434 be tampered with or bypassed, and whether TSFs that provide security domains isolate those
 4435 domains from each other.

4436 **11.3.1.2 Input**

4437 The evaluation evidence for this sub-activity is:

- 4438 a) the ST;
- 4439 b) the functional specification;
- 4440 c) the TOE design;
- 4441 d) the security architecture description;
- 4442 e) the implementation representation (if available);
- 4443 f) the operational user guidance;

4444 **11.3.1.3 Application notes**

4445 The notions of self-protection, domain separation, and non-bypassability are distinct from security
 4446 functionality expressed in ISO/IEC 15408-2 SFRs because self-protection and non-bypassability
 4447 largely have no directly observable interface at the TSF. Rather, they are properties of the TSF that
 4448 are achieved through the design of the TOE, and enforced by the correct implementation of that
 4449 design. Also, the evaluation of these properties is less straight-forward than the evaluation of
 4450 mechanisms; it is more difficult to check for the absence of functionality than for its presence.
 4451 However, the determination that these properties are being satisfied is just as critical as the
 4452 determination that the mechanisms are properly implemented.

4453 The overall approach used is that the developer provides a TSF that meets the above-mentioned
 4454 properties, and provides evidence (in the form of documentation) that can be analysed to show
 4455 that the properties are indeed met. The evaluator has the responsibility for looking at the evidence
 4456 and, coupled with other evidence delivered for the TOE, determining that the properties are
 4457 achieved. The work units can be characterised as those detailing with what information has to be
 4458 provided, and those dealing with the actual analysis the evaluator performs.

4459 The security architecture description describes how domains are defined and how the TSF keeps
 4460 them separate. It describes what prevents untrusted processes from getting to the TSF and
 4461 modifying it. It describes what ensures that all resources under the TSF's control are adequately
 4462 protected and that all actions related to the SFRs are mediated by the TSF. It explains any role the
 4463 environment plays in any of these (e.g. presuming it gets correctly invoked by its underlying
 4464 environment, how is its security functionality invoked?). In short, it explains how the TOE is
 4465 considered to be providing any kind of *security* service.

4466 The analyses the evaluator performs must be done in the context of all of the development
 4467 evidence provided for the TOE, at the level of detail the evidence is provided. At lower assurance
 4468 levels, there should not be the expectation that, for example, TSF self-protection is completely
 4469 analysed, because only high-level design representations will be available. The evaluator also
 4470 needs to be sure to use information gleaned from other portions of their analysis (e.g., analysis of
 4471 the TOE design) in making their assessments for the properties being examined in the following
 4472 work units.

4473 **11.3.1.4 Action ADV_ARC.1.1E**

4474 ISO/IEC 15408-3 ADV_ARC.1.1C: *The security architecture description shall be at a level of detail*
 4475 *commensurate with the description of the SFR-enforcing abstractions described in the TOE design*
 4476 *document.*

4477 **11.3.1.4.1 Work unit ADV_ARC.1-1**

4478 The evaluator ***shall examine*** the security architecture description to determine that the
 4479 information provided in the evidence is presented at a level of detail commensurate with the
 4480 descriptions of the SFR-enforcing abstractions contained in the functional specification and TOE
 4481 design document.

4482 With respect to the functional specification, the evaluator should ensure that the self-protection
 4483 functionality described cover those effects that are evident at the TSFI. Such a description might
 4484 include protection placed upon the executable images of the TSF, and protection placed on objects
 4485 (e.g., files used by the TSF). The evaluator ensures that the functionality that might be invoked
 4486 through the TSFI is described.

4487 If Evaluation of sub-activity (ADV_TDS.1) or Evaluation of sub-activity (ADV_TDS.2) is included, the
 4488 evaluator ensures the security architecture description contains information on how any
 4489 subsystems that contribute to TSF domain separation work.

4490 If Evaluation of sub-activity (ADV_TDS.3) or higher is available, the evaluator ensures that the
 4491 security architecture description also contains implementation-dependent information. For
 4492 example, such a description might contain information pertaining to coding conventions for
 4493 parameter checking that would prevent TSF compromises (e.g. buffer overflows), and information
 4494 on stack management for call and return operations. The evaluator checks the descriptions of the
 4495 mechanisms to ensure that the level of detail is such that there is little ambiguity between the
 4496 description in the security architecture description and the implementation representation.

4497 The evaluator action related to this work unit is assigned a fail verdict if the security architecture
 4498 description mentions any module, subsystem, or interface that is not described in the functional
 4499 specification or TOE design document.

4500 ISO/IEC 15408-3 ADV_ARC.1.2C: *The security architecture description shall describe the security*
 4501 *domains maintained by the TSF consistently with the SFRs.*

4502 **11.3.1.4.2 Work unit ADV_ARC.1-2**

4503 The evaluator ***shall examine*** the security architecture description to determine that it describes
 4504 the security domains maintained by the TSF.

4505 Security domains refer to environments supplied by the TSF for use by potentially-harmful
 4506 entities; for example, a typical secure operating system supplies a set of resources (address space,
 4507 per-process environment variables) for use by processes with limited access rights and security
 4508 properties. The evaluator determines that the developer's description of the security domains
 4509 takes into account all of the SFRs claimed by the TOE.

4510 For some TOEs such domains do not exist because all of the interactions available to users are
 4511 severely constrained by the TSF. A packet-filter firewall is an example of such a TOE. Users on the
 4512 LAN or WAN do not interact with the TOE, so there need be no security domains; there are only
 4513 data structures maintained by the TSF to keep the users' packets separated. The evaluator ensures
 4514 that any claim that there are no domains is supported by the evidence and that no such domains
 4515 are, in fact, available.

4516 ISO/IEC 15408-3 ADV_ARC.1.3C: *The security architecture description shall describe how the TSF*
 4517 *initialisation process is secure.*

4518 **11.3.1.4.3 Work unit ADV_ARC.1-3**

4519 The evaluator ***shall examine*** the security architecture description to determine that the
 4520 initialisation process preserves security.

4521 The information provided in the security architecture description relating to TSF initialisation is
 4522 directed at the TOE components that are involved in bringing the TSF into an initial secure state
 4523 (i.e. when all parts of the TSF are operational) when power-on or a reset is applied. This discussion
 4524 in the security architecture description should list the system initialisation components and the
 4525 processing that occurs in transitioning from the "down" state to the initial secure state.

4526 It is often the case that the components that perform this initialisation function are not accessible
 4527 after the secure state is achieved; if this is the case then the security architecture description
 4528 identifies the components and explains how they are not reachable by untrusted entities after the
 4529 TSF has been established. In this respect, the property that needs to be preserved is that these
 4530 components either 1) cannot be accessed by untrusted entities after the secure state is achieved, or
 4531 2) if they provide interfaces to untrusted entities, these TSFI cannot be used to tamper with the
 4532 TSF.

4533 The TOE components related to TSF initialisation, then, are treated themselves as part of the TSF,
 4534 and analysed from that perspective. It should be noted that even though these are treated as part of
 4535 the TSF, it is likely that a justification (as allowed by

4536 **Objectives**

4537 The objective of this sub-activity is to determine that the implementation representation made
 4538 available by the developer is suitable for use in other analysis activities; suitability is judged by its
 4539 conformance to the requirements for this component.

4540 **Input**

4541 The evaluation evidence for this sub-activity is:

4542

4543 the implementation representation;

4544 the documentation of the development tools, as resulting from ALC_TAT;

4545 the TOE design description.

4546 **Application notes**

4547 The entire implementation representation is made available to ensure that analysis activities are
4548 not curtailed due to lack of information. This does not, however, imply that all of the representation
4549 is examined in detail when the analysis activities are being performed. This is likely impractical in
4550 almost all cases, in addition to the fact that it most likely will not result in a higher-assurance TOE.

4551 The new aspect for ADV_IMP.2 in comparison to ADV_IMP.1 is that the developer needs to
4552 demonstrate and the evaluator will confirm that the complete implementation representation is
4553 mapped to the TOE design description. This does, however, not imply that all other work units
4554 need an examination of the complete implementation representation. Aspects like appropriate
4555 level of detail and form of the implementation representation can be covered by sampling as for
4556 ADV_IMP.1.

4557 **Action ADV_IMP.2.1E**

4558 ISO/IEC 15408-3 ADV_IMP.2.1C *The implementation representation shall define the TSF to*
4559 *a level of detail such that the TSF can be generated without further design decisions.*

4560 **Work unit ADV_IMP.2-1**

4561 The evaluator **shall check** that the implementation representation defines the TSF to a level of
4562 detail such that the TSF can be generated without further design decisions.

4563 Source code or hardware diagrams and/or IC hardware design language code or layout data that
4564 are used to build the actual hardware are examples of parts of an implementation representation.
4565 The evaluator samples the implementation representation to gain confidence that it is at the
4566 appropriate level and not, for instance, a pseudo-code level which requires additional design
4567 decisions to be made. The evaluator is encouraged to perform a quick check when first looking at
4568 the implementation representation to assure themselves that the developer is on the right track.
4569 However, the evaluator is also encourage to perform the bulk of this check while working on other
4570 work units that call for examining the implementation; this will ensure the sample examined for
4571 this work unit is relevant.

4572 If the evaluator has the possibility to actually execute or witness the "built" procedure used to
4573 transfer the implementation representation into the actual implementation, and to compare the
4574 result to the TOE as delivered, this may provide an easier and at the same time more reliable check
4575 for this work unit (and possibly also for the following one).

4576 ISO/IEC 15408-3 ADV_IMP.2.2C *The implementation representation shall be in the form*
4577 *used by the development personnel.*

4578 **Work unit ADV_IMP.2-2**

4579 The evaluator **shall check** that the implementation representation is in the form used by
4580 development personnel.

4581 The implementation representation is manipulated by the developer in form that it suitable for
4582 transformation to the actual implementation. For instance, the developer may work with files
4583 containing source code, which is eventually compiled to become part of the TSF. The developer
4584 makes available the implementation representation in the form they use, so that the evaluator may
4585 use automated techniques in the analysis. This also increases the confidence that the
4586 implementation representation examined is actually the one used in the production of the TSF (as
4587 opposed to the case where it is supplied in an alternate presentation format, such as a word
4588 processor document). It should be noted that other forms of the implementation representation
4589 may also be used by the developer; these forms are supplied as well. The overall goal is to supply
4590 the evaluator with the information that will maximise the evaluator's analysis efforts.

- 4591 The evaluator samples the implementation representation to gain confidence that it is the version
 4592 that is usable by the developer. The sample is such that the evaluator has assurance that all areas of
 4593 the implementation representation are in conformance with the requirement; however, a complete
 4594 examination of the entire implementation representation is unnecessary.
- 4595 Conventions in some forms of the implementation representation may make it difficult or
 4596 impossible to determine from just the implementation representation itself what the actual result
 4597 of the compilation or run-time interpretation will be. For example, compiler directives for C
 4598 language compilers will cause the compiler to exclude or include entire portions of the code.
- 4599 Some forms of the implementation representation may require additional information because
 4600 they introduce significant barriers to understanding and analysis. Examples include shrouded
 4601 source code or source code that has been obfuscated in other ways such that it prevents
 4602 understanding and/or analysis. These forms of implementation representation typically result
 4603 from by taking a version of the implementation representation that is used by the TOE developer
 4604 and running a shrouding or obfuscation program on it. While the shrouded representation is what
 4605 is compiled and may be closer to the implementation (in terms of structure) than the original, un-
 4606 shrouded representation, supplying such obfuscated code may cause significantly more time to be
 4607 spent in analysis tasks involving the representation. When such forms of representation are
 4608 created, the components require details on the shrouding tools/algorithms used so that the un-
 4609 shrouded representation can be supplied, and the additional information can be used to gain
 4610 confidence that the shrouding process does not compromise any security mechanisms.
- 4611 The evaluator samples the implementation representation to gain confidence that all of the
 4612 information needed to interpret the implementation representation has been supplied. Note that
 4613 the tools are among those referenced by Tools and techniques (ALC_TAT) components. The
 4614 evaluator is encouraged to perform a quick check when first looking at the implementation
 4615 representation to assure themselves that the developer is on the right track. However, the
 4616 evaluator is also encouraged to perform the bulk of this check while working on other work units
 4617 that call for examining the implementation; this will ensure the sample examined for this work unit
 4618 is relevant.
- 4619 ISO/IEC 15408-3 ADV_IMP.2.3C *The mapping between the TOE design description and the*
 4620 *entire implementation representation shall demonstrate their correspondence.*
- 4621 **Work unit ADV_IMP.2-3**
- 4622 The evaluator ***shall examine*** the mapping between the TOE design description and the entire
 4623 implementation representation to determine that it is accurate.
- 4624 The evaluator augments the determination of existence (specified in work unit ADV_IMP.2-1) by
 4625 verifying the accuracy of the implementation representation and the TOE design description. For
 4626 those parts of TOE design description that are interesting, the evaluator would verify the
 4627 implementation representation accurately reflects the description provided in the TOE design
 4628 description.
- 4629 For example, the TOE design description might identify a login module that is used to identify and
 4630 authenticate users. If user authentication is sufficiently significant, the evaluator would verify that
 4631 the corresponding code in fact implements that service as described in the TOE design description.
 4632 It might also be worthwhile to verify that the code accepts the parameters as described in the
 4633 functional specification.
- 4634 Usually it will be expected that the evaluator considers at least the functionality required by the
 4635 SFRs chosen in the ST and aspects described in the security architecture description as
 4636 "interesting" in the sense discussed above. Note however that not all aspects of the security
 4637 architecture are necessarily traceable to specific parts of the implementation representation.

4638 It is worth pointing out the developer must perform the mapping for the entire implementation
4639 representation, thereby guaranteeing that the chosen sample will be covered.

4640 **Work unit ADV_IMP.2-4**

4641 The evaluator ***shall examine*** the mapping between the TOE design description and the entire
4642 implementation representation to determine that it is complete.

4643 Note that the completeness here is relevant in both directions: The complete TOE design needs to
4644 be covered by the implementation representation and all parts of the implementation
4645 representation needs to be mapped to a corresponding part of the TOE design.

4646 In order to confirm that the entire implementation representation is covered by the mapping the
4647 evaluator will not need to examine the content of every part of the implementation representation.
4648 If (in the case of a software TOE) the mapping is for example described by mapping each source
4649 code file to a module in the TOE design description, it will be sufficient if this mapping is plausible
4650 from the role of the source code file the evaluator can conclude from information like the naming of
4651 the source code files, their grouping in subdirectories or their grouping in "built" procedures. Note,
4652 that aspects of accuracy are covered by the preceding work unit.

4653 In order to confirm that the entire design description is covered by the implementation, the
4654 evaluator may either use a similar argument as in the other direction, i. e. that all modules
4655 contained in the TOE design description are mapped to parts of the implementation representation
4656 in a plausible way. In addition, if the evaluator has established in the preceding work unit that all
4657 SFRs and all applicable parts of the security architecture description are traceable to the
4658 implementation representation this may be seen as sufficient evidence that the mapping is
4659 complete.

4660 TSF internals (ADV_INT)) can be made that they do not have to meet the internal structuring
4661 requirements of ADV_INT.

4662 ISO/IEC 15408-3 ADV_ARC.1.4C: *The security architecture description shall demonstrate that the*
4663 *TSF protects itself from tampering.*

4664 **11.3.1.8.5 Work unit ADV_ARC.1-4**

4665 The evaluator ***shall examine*** the security architecture description to determine that it contains
4666 information sufficient to support a determination that the TSF is able to protect itself from
4667 tampering by untrusted active entities.

4668 "Self-protection" refers to the ability of the TSF to protect itself from manipulation from external
4669 entities that may result in changes to the TSF. For TOEs that have dependencies on other IT entities,
4670 it is often the case that the TOE uses services supplied by the other IT entities in order to perform
4671 its functions. In such cases, the TSF alone does not protect itself because it depends on the other IT
4672 entities to provide some of the protection. For the purposes of the security architecture description,
4673 the notion of *self-protection* applies only to the services provided by the TSF through its TSFI, and
4674 not to services provided by underlying IT entities that it uses.

4675 Self-protection is typically achieved by a variety of means, ranging from physical and logical
4676 restrictions on access to the TOE; to hardware-based means (e.g. "execution rings" and memory
4677 management functionality); to software-based means (e.g. boundary checking of inputs on a
4678 trusted server). The evaluator determines that all such mechanisms are described.

4679 The evaluator determines that the design description covers how user input is handled by the TSF
4680 in such a way that the TSF does not subject itself to being corrupted by that user input. For example,
4681 the TSF might implement the notion of privilege and protect itself by using privileged-mode
4682 routines to handle user input. The TSF might make use of processor-based separation mechanisms
4683 such as privilege levels or rings. The TSF might implement software protection constructs or

4684 coding conventions that contribute to implementing separation of software domains, perhaps by
 4685 delineating user address space from system address space. And the TSF might have reliance its
 4686 environment to provide some support to the protection of the TSF.

4687 All of the mechanisms contributing to the domain separation functions are described. The
 4688 evaluator should use knowledge gained from other evidence (functional specification, TOE design,
 4689 TSF internals description, other parts of the security architecture description, or implementation
 4690 representation, as included in the assurance package for the TOE) in determining if any
 4691 functionality contributing to self-protection was described that is not present in the security
 4692 architecture description.

4693 Accuracy of the description of the self-protection mechanisms is the property that the description
 4694 faithfully describes what is implemented. The evaluator should use other evidence (functional
 4695 specification, TOE design, TSF Internals documentation, other parts of the security architecture
 4696 description, implementation representation, as included in the ST for the TOE) in determining
 4697 whether there are discrepancies in any descriptions of the self-protection mechanisms. If
 4698 Implementation representation (ADV_IMP) is included in the assurance package for the TOE, the
 4699 evaluator will choose a sample of the implementation representation; the evaluator should also
 4700 ensure that the descriptions are accurate for the sample chosen. If an evaluator cannot understand
 4701 how a certain self-protection mechanism works or could work in the system architecture, it may be
 4702 the case that the description is not accurate.

4703 ISO/IEC 15408-3 ADV_ARC.1.5C: *The security architecture description shall demonstrate that the*
 4704 *TSF prevents bypass of the SFR-enforcing functionality.*

4705 **11.3.1.8.6 Work unit ADV_ARC.1-5**

4706 The evaluator ***shall examine*** the security architecture description to determine that it presents an
 4707 analysis that adequately describes how the SFR-enforcing mechanisms cannot be bypassed.

4708 Non-bypassability is a property that the security functionality of the TSF (as specified by the SFRs)
 4709 is always invoked. For example, if access control to files is specified as a capability of the TSF via an
 4710 SFR, there must be no interfaces through which files can be accessed without invoking the TSF's
 4711 access control mechanism (such as an interface through which a raw disk access takes place).

4712 Describing how the TSF mechanisms cannot be bypassed generally requires a systematic argument
 4713 based on the TSF and the TSFIs. The description of how the TSF works (contained in the design
 4714 decomposition evidence, such as the functional specification, TOE design documentation) - along
 4715 with the information in the TSS - provides the background necessary for the evaluator to
 4716 understand what resources are being protected and what security functions are being provided.
 4717 The functional specification provides descriptions of the TSFIs through which the
 4718 resources/functions are accessed.

4719 The evaluator assesses the description provided (and other information provided by the developer,
 4720 such as the functional specification) to ensure that no available interface can be used to bypass the
 4721 TSF. This means that every available interface must be either unrelated to the SFRs that are
 4722 claimed in the ST (and does not interact with anything that is used to satisfy SFRs) or else uses the
 4723 security functionality that is described in other development evidence in the manner described.
 4724 For example, a game would likely be unrelated to the SFRs, so there must be an explanation of how
 4725 it cannot affect security. Access to user data, however, is likely to be related to access control SFRs,
 4726 so the explanation would describe how the security functionality works when invoked through the
 4727 data-access interfaces. Such a description is needed for every available interface.

4728 An example of a description follows. Suppose the TSF provides file protection. Further suppose that
 4729 although the "traditional" system call TSFIs for open, read, and write invoke the file protection
 4730 mechanism described in the TOE design, there exists a TSFI that allows access to a batch job facility
 4731 (creating batch jobs, deleting jobs, modifying unprocessed jobs). The evaluator should be able to
 4732 determine from the vendor-provided description that this TSFI invokes the same protection

4733 mechanisms as do the “traditional” interfaces. This could be done, for example, by referencing the
 4734 appropriate subclauses of the TOE design that discuss *how* the batch job facility TSFI achieves its
 4735 security objectives.

4736 Using this same example, suppose there is a TSFI whose sole purpose is to display the time of day.
 4737 The evaluator should determine that the description adequately argues that this TSFI is not
 4738 capable of manipulating any protected resources and should not invoke any security functionality.

4739 Another example of bypass is when the TSF is supposed to maintain confidentiality of a
 4740 cryptographic key (one is allowed to use it for cryptographic operations, but is not allowed to
 4741 read/write it). If an attacker has direct physical access to the device, they might be able to examine
 4742 side-channels such as the power usage of the device, the exact timing of the device, or even any
 4743 electromagnetic emanations of the device and, from this, infer the key.

4744 If such side-channels may be present, the demonstration should address the mechanisms that
 4745 prevent these side-channels from occurring, such as random internal clocks, dual-line technology
 4746 etc. Verification of these mechanisms would be verified by a combination of purely design-based
 4747 arguments and testing.

4748 For a final example using security functionality rather than a protected resource, consider an ST
 4749 that contains **FCO_NRO.2 Enforced proof of origin**, which requires that the TSF provides evidence
 4750 of origination for information types specified in the ST. Suppose that the “information types”
 4751 included all information that is sent by the TOE via e-mail. In this case, the evaluator should
 4752 examine the description to ensure that all TSFI that can be invoked to send e-mail perform the
 4753 “evidence of origination generation” function are detailed. The description might point to user
 4754 guidance to show all places where e-mail can originate (e.g., e-mail program, notification from
 4755 scripts/batch jobs) and then how each of these places invokes the evidence generation function.

4756 The evaluator should also ensure that the description is comprehensive, in that each interface is
 4757 analysed with respect to the entire set of claimed SFRs. This may require the evaluator to examine
 4758 supporting information (functional specification, TOE design, other parts of the security
 4759 architecture description, operational user guidance, and perhaps even the implementation
 4760 representation, as provided for the TOE) to determine that the description has correctly capture all
 4761 aspects of an interface. The evaluator should consider what SFRs each TSFI might affect (from the
 4762 description of the TSFI and its implementation in the supporting documentation), and then
 4763 examine the description to determine whether it covers those aspects.

4764 **11.4 Functional specification (ADV_FSP)**

4765 **11.4.1 Evaluation of sub-activity (ADV_FSP.1)**

4766 **11.4.1.1 Objectives**

4767 The objective of this sub-activity is to determine whether the developer has provided a high-level
 4768 description of at least the SFR-enforcing and SFR-supporting TSFIs, in terms of descriptions of their
 4769 parameters. There is no other required evidence that can be expected to be available to measure
 4770 the accuracy of these descriptions; the evaluator merely ensures the descriptions seem plausible.

4771 **11.4.1.2 Input**

4772 The evaluation evidence for this sub-activity is:

- 4773 a) the ST;
- 4774 b) the functional specification;
- 4775 c) the operational user guidance;

4776 **11.4.1.3 Action ADV_FSP.1.1E**

4777 ISO/IEC 15408-3 ADV_FSP.1.1C: *The functional specification shall describe the purpose and method*
 4778 *of use for each SFR-enforcing and SFR-supporting TSFI.*

4779 **11.4.1.3.1 Work unit ADV_FSP.1-1**

4780 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of
 4781 each SFR-supporting and SFR-enforcing TSFI.

4782 The purpose of a TSFI is a general statement summarising the functionality provided by the
 4783 interface. It is not intended to be a complete statement of the actions and results related to the
 4784 interface, but rather a statement to help the reader understand in general what the interface is
 4785 intended to be used for. The evaluator should not only determine that the purpose exists, but also
 4786 that it accurately reflects the TSFI by taking into account other information about the interface,
 4787 such as the description of the parameters; this can be done in association with other work units for
 4788 this component.

4789 If an action available through an interface plays a role in enforcing any security policy on the TOE
 4790 (that is, if one of the actions of the interface can be traced to one of the SFRs levied on the TSF),
 4791 then that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but
 4792 also refer to any functionality specified by one of the SFRs contained in the ST. Note that it is
 4793 possible that an interface may have various actions and results, some of which may be SFR-
 4794 enforcing and some of which may not.

4795 Interfaces to (or actions available through an interface relating to) actions that SFR-enforcing
 4796 functionality depends on, but need only to function correctly in order for the security policies of
 4797 the TOE to be preserved, are termed *SFR supporting*. Interfaces to actions on which SFR-enforcing
 4798 functionality has no dependence are termed *SFR non-interfering*.

4799 It should be noted that in order for an interface to be SFR supporting or SFR non-interfering it must
 4800 have *no* SFR-enforcing actions or results. In contrast, an SFR-enforcing interface may have SFR-
 4801 supporting actions (for example, the ability to set the system clock may be an SFR-enforcing action
 4802 of an interface, but if that same interface is used to display the system date that action may only be
 4803 SFR supporting). An example of a purely SFR-supporting interface is a system call interface that is
 4804 used both by untrusted users and by a portion of the TSF that is running in user mode.

4805 At this level, it is unlikely that a developer will have expended effort to label interfaces as SFR-
 4806 enforcing and SFR-supporting. In the case that this has been done, the evaluator should verify to
 4807 the extent that supporting documentation (e.g., operational user guidance) allows that this
 4808 identification is correct. Note that this identification activity is necessary for several work units for
 4809 this component.

4810 In the more likely case that the developer has not labelled the interfaces, the evaluator must
 4811 perform their own identification of the interfaces first, and then determine whether the required
 4812 information (for this work unit, the purpose) is present. Again, because of the lack of supporting
 4813 evidence this identification will be difficult and have low assurance that all appropriate interfaces
 4814 have been correctly identified, but nonetheless the evaluator examines other evidence available for
 4815 the TOE to ensure as complete coverage as is possible.

4816 **11.4.1.3.2 Work unit ADV_FSP.1-2**

4817 The evaluator ***shall examine*** the functional specification to determine that the method of use for
 4818 each SFR-supporting and SFR-enforcing TSFI is given.

4819 See work unit ADV_FSP.1-1 for a discussion on the identification of SFR-supporting and SFR-
 4820 enforcing TSFI.

4821 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the
 4822 actions and obtain the results associated with the TSFI. The evaluator should be able to determine,
 4823 from reading this material in the functional specification, how to use each interface. This does not
 4824 necessarily mean that there needs to be a separate method of use for each TSFI, as it may be
 4825 possible to describe in general how kernel calls are invoked, for instance, and then identify each
 4826 interface using that general style. Different types of interfaces will require different method of use
 4827 specifications. APIs, network protocol interfaces, system configuration parameters, and hardware
 4828 bus interfaces all have very different methods of use, and this should be taken into account by the
 4829 developer when developing the functional specification, as well as by the evaluator evaluating the
 4830 functional specification.

4831 For administrative interfaces, whose functionality is documented as being inaccessible to
 4832 untrusted users, the evaluator ensures that the method of making the functions inaccessible is
 4833 described in the functional specification. It should be noted that this inaccessibility needs to be
 4834 tested by the developer in their test suite.

4835 ISO/IEC 15408-3 ADV_FSP.1.2C: *The functional specification shall identify all parameters associated*
 4836 *with each SFR-enforcing and SFR-supporting TSFI.*

4837 **11.4.1.3.3 Work unit ADV_FSP.1-3**

4838 The evaluator ***shall examine*** the presentation of the TSFI to determine that it identifies all
 4839 parameters associated with each SFR-enforcing and SFR-supporting TSFI.

4840 See work unit ADV_FSP.1-1 for a discussion on the identification of SFR-supporting and SFR-
 4841 enforcing TSFI.

4842 The evaluator examines the functional specification to ensure that all of the parameters are
 4843 described for identified TSFI. Parameters are explicit inputs or outputs to an interface that control
 4844 the behaviour of that interface. For examples, parameters are the arguments supplied to an API;
 4845 the various fields in packet for a given network protocol; the individual key values in the Windows
 4846 Registry; the signals across a set of pins on a chip; etc.

4847 While difficult to obtain much assurance that all parameters for the applicable TSFI have been
 4848 identified, the evaluator should also check other evidence provided for the evaluation (e.g.,
 4849 operational user guidance) to see if behaviour or additional parameters are described there but not
 4850 in the functional specification.

4851 ISO/IEC 15408-3 ADV_FSP.1.3C: *The functional specification shall provide rationale for the implicit*
 4852 *categorisation of interfaces as SFR-non-interfering.*

4853 **11.4.1.3.4 Work unit ADV_FSP.1-4**

4854 The evaluator ***shall examine*** the rationale provided by the developer for the implicit
 4855 categorisation of interfaces as SFR-non-interfering to determine that it is accurate.

4856 In the case where the developer has provided adequate documentation to perform the analysis
 4857 called for by the rest of the work units for this component without explicitly identifying SFR-
 4858 enforcing and SFR-supporting interfaces, this work unit should be considered satisfied.

4859 This work unit is intended to apply to cases where the developer has not described a portion of the
 4860 TSFI, claiming that it is SFR-non-interfering and therefore not subject to other requirements of this
 4861 component. In such a case, the developer provides a rationale for this characterisation in sufficient
 4862 detail such that the evaluator understands the rationale, the characteristics of the interfaces
 4863 affected (e.g., their high-level function with respect to the TOE, such as “colour palette
 4864 manipulation”), and that the claim that these are SFR-non-interfering is supported. Given the level
 4865 of assurance the evaluator should not expect more detail than is provided for the SFR-enforcing or

4866 SFR-supporting interfaces, and in fact the detail should be much less. In most cases, individual
4867 interfaces should not need to be addressed in the developer-provided rationale subclause.

4868 ISO/IEC 15408-3 ADV_FSP.1.4C: *The tracing shall demonstrate that the SFRs trace to TSFIs in the*
4869 *functional specification.*

4870 **11.4.1.3.5 Work unit ADV_FSP.1-5**

4871 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

4872 The tracing is provided by the developer to serve as a guide to which SFRs are related to which
4873 TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the
4874 following work units, in which the evaluator verifies its completeness and accuracy.

4875 **11.4.1.4 Action ADV_FSP.1.2E**

4876 **11.4.1.4.1 Work unit ADV_FSP.1-6**

4877 The evaluator ***shall examine*** the functional specification to determine that it is a complete
4878 instantiation of the SFRs.

4879 To ensure that all SFRs are covered by the functional specification, as well as the test coverage
4880 analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.1-5 a map between
4881 the TOE security functional requirements and the TSFI). Note that this map may have to be at a
4882 level of detail below the component or even element level of the requirements, because of
4883 operations (assignments, refinements, selections) performed on the functional requirement by the
4884 ST author.

4885 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained,
4886 for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three
4887 different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and
4888 claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to
4889 TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper
4890 interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of
4891 parameters for a given interface.

4892 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
4893 boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the
4894 TSFI. The analysis for those requirements will be performed in the analysis for the TOE design
4895 (ADV_TDS) when included in the ST. It is also important to note that since the parameters
4896 associated with TSFIs must be fully specified, the evaluator should be able to determine if all
4897 aspects of an SFR appear to be implemented at the interface level.

4898 **11.4.1.4.2 Work unit ADV_FSP.1-7**

4899 The evaluator ***shall examine*** the functional specification to determine that it is an accurate
4900 instantiation of the SFRs.

4901 For each functional requirement in the ST that results in effects visible at the TSF boundary, the
4902 information in the associated TSFI for that requirement specifies the required functionality
4903 described by the requirement. For example, if the ST contains a requirement for access control lists,
4904 and the only TSFI that map to that requirement specify functionality for Unix-style protection bits,
4905 then the functional specification is not accurate with respect to the requirements.

4906 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
4907 boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements
4908 to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE
4909 design (ADV_TDS) when included in the ST.

4910 **11.4.2 Evaluation of sub-activity (ADV_FSP.2)**

4911 **11.4.2.1 Objectives**

4912 The objective of this sub-activity is to determine whether the developer has provided a description
4913 of the TSFIs in terms of their purpose, method of use, and parameters. In addition, the SFR-
4914 enforcing actions, results and error messages of each TSFI that is SFR-enforcing are also described.

4915 **11.4.2.2 Input**

4916 The evaluation evidence for this sub-activity that is required by the work-units is:

- 4917 a) the ST;
- 4918 b) the functional specification;
- 4919 c) the TOE design.

4920 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

- 4921 a) the security architecture description;
- 4922 b) the operational user guidance;

4923 **11.4.2.3 Action ADV_FSP.2.1E**

4924 ISO/IEC 15408-3 ADV_FSP.2.1C: *The functional specification shall completely represent the TSF.*

4925 **11.4.2.3.1 Work unit ADV_FSP.2-1**

4926 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully
4927 represented.

4928 **11.4.2.4 The identification of the TSFI is a necessary prerequisite to all other activities in**
4929 **this sub-activity. The TSF must be identified (done as part of the Objectives)**

4930 **11.4.2.4** The objectives of this sub-activity are to determine whether the formal security policy model of
4931 the TSF clearly and consistently describes the rules and characteristics of the security policies
4932 and whether this description corresponds with the description of security functions in the
4933 functional specification.

4934 **11.4.2.4 Input**

4935 **11.4.2.4** The evaluation evidence for this sub-activity is:

- 4936 **11.4.2.4** the ST;
- 4937 **11.4.2.4** the functional specification;
- 4938 **11.4.2.4** formal security policy model (ADV_SPM.1.1D);
- 4939 **11.4.2.4** formal proof of correspondence between the model and any formal functional specification
4940 (ADV_SPM.1.3D);
- 4941 **11.4.2.4** demonstration of correspondence between the model and the functional specification
4942 (ADV_SPM.1.4D).

4943 **11.4.2.4 Application notes**

4944 **11.4.2.4** This activity applies to cases where the developer has provided a formal security policy
4945 model of the TOE.

4946 **11.4.2.4** A formal TOE security policy model is a representation of the rules (synonymously
4947 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
4948 terms. Their formal counterparts are called security properties and security features,
4949 respectively. The representation includes but is not limited to algebraic specifications, finite state
4950 machines and logic formalisms strong enough to formally infer the properties from the features.
4951 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
4952 characteristics are mapped to the respective properties and features.

4953 **11.4.2.4** The creation of a formal security policy model helps to identify and eliminate
4954 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
4955 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
4956 judgement of how well the developer has understood the security functionality being
4957 implemented and whether there are inconsistencies between the security requirements and the
4958 TOE design. The confidence in the model is accompanied by a proof that it contains no
4959 inconsistencies.

4960 **11.4.2.4** A formal security model is a precise formal presentation of the important aspects of
4961 security and their relationship to the behaviour of the TOE; it identifies the set of rules
4962 (principles) that defines the TOE security policy and the set of practises (characteristics) that
4963 regulates how the TSF manages, protects, and otherwise controls the system resources. The
4964 model includes the set of restrictions and properties that specify how information and computing
4965 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
4966 engineering arguments showing that these restrictions and properties play a key role in the
4967 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
4968 as well as ancillary text to explain the model and to provide it with context. The security
4969 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
4970 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

4971 **11.4.2.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
4972 considering the proposed security requirements of the ST. The informal abstraction is taken to be
4973 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
4974 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
4975 are always prone to fallacies; especially if relationships among subjects, objects and operations
4976 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
4977 characteristics of the security policy model are mapped to respective properties and features
4978 within some formal system, whose rigour and strength can afterwards be used to obtain the
4979 security properties by means of theorems and formal proof.

4980 **11.4.2.4** While the term “formal security policy model” is used in academic circles, the CC's
4981 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
4982 claimed. Therefore, the formal security policy model is merely a formal representation of the set
4983 of SFRs being claimed.

4984 **11.4.2.4** The term security policy has traditionally been associated with only access control
4985 policies, whether label-based (mandatory access control) or user-based (discretionary access
4986 control). However, a security policy is not limited to access control; there are also audit policies,
4987 identification policies, authentication policies, encryption policies, management policies, and any
4988 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
4989 contains an assignment for identifying these policies that are formally modelled.

4990 **11.4.2.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
4991 because either a given policy can not be formally modelled in the otherwise well suited

- 4992 framework, or because the nature of the TOE renders impossible the modelling of policies that
4993 would otherwise be possible to model.
- 4994 **11.4.2.4 Action ADV_SPM.1.1E**
- 4995 **11.4.2.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
4996 *text as required, and identify the security policies of the TSF that are modelled.*
- 4997 **11.4.2.4 Work unit ADV_SPM.1-1**
- 4998 **11.4.2.4** The evaluator ***shall examine the TOE security policy model to determine that it is***
4999 ***written in a formal style.***
- 5000 **11.4.2.4** The evaluator identifies the formal framework upon which the TOE security policy
5001 model is based and ensures that it is founded on well established mathematical concepts. **They**
5002 **also identify the security properties and features addressed in the application notes and ensure**
5003 **the formalization of at least one security policy.**
- 5004 **11.4.2.4** For guidance on formal methods refer to ISO/IEC 15408-3
- 5005 **11.4.2.4 Work unit ADV_SPM.1-2**
- 5006 **11.4.2.4** The evaluator ***shall examine the TOE security policy model to determine that it***
5007 ***contains all necessary informal explanatory text.***
- 5008 **11.4.2.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
5009 to make clear the meaning of any formal notation and how they are used) including the security
5010 properties and features.
- 5011 **11.4.2.4 Work unit ADV_SPM.1-3**
- 5012 **11.4.2.4** The evaluator ***shall examine the TOE security policy model to determine that all***
5013 ***security policies of the TSF are identified that are modelled.***
- 5014 **11.4.2.4** The evaluator determines whether the SPM identifies the security policies for which a
5015 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
5016 of the modelled policies.
- 5017 **11.4.2.4** The evaluator determines whether the list of security policies identified by the SPM is
5018 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 5019 **11.4.2.4** The evaluator determines whether for each security policy identified by the SPM a
5020 model is in fact provided.
- 5021 **11.4.2.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
5022 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
5023 *not secure.*
- 5024 **11.4.2.4 Work unit ADV_SPM.1-4**
- 5025 **11.4.2.4** The evaluator ***shall examine the principles and characteristics of the security policies***
5026 ***to determine that the modelled security behaviour of the TOE is clearly articulated.***
- 5027 **11.4.2.4** The security policies are expressed in terms of security principles (rules) which are
5028 modelled by security properties and define the secure state of the TOE. For example, a model
5029 based on state transitions could describe the security policies in terms of principles of its states,
5030 identify its initial state, and define what it means to be a secure state.

5031 5032	11.4.2.4 The evaluator determines that the security policies are reflected within their formal counterparts of the TSP model.
5033 5034 5035 5036 5037 5038	11.4.2.4 The TOE security behaviour is expressed in terms of security characteristics (i.e. portions of TOE security functionality managing, protecting, and otherwise controlling the system resources including attributes and conditions of the TOE) which are modelled by security features. For example, a model based on state transitions could describe the characteristics as possible actions in each secure state in a level of detail sufficient to decide into which state the TOE will be transformed by that action.
5039 5040	11.4.2.4 Together the security principles and characteristics describe the entire security posture of the TOE.
5041 5042 5043 5044 5045 5046	11.4.2.4 In the context of a formal TOE security policy model the security behaviour is considered to be clearly articulated only if an adequate mapping from principles and characteristics to their respective formal counterparts properties and features has been given. The mapping is considered to be adequate if the level of abstraction from the TOE's realization is detailed enough to allow for correct identification of all security objectives and the relation to the security environment.
5047 5048 5049	11.4.2.4 The above condition for clear articulation is necessary but not sufficient. An informal interpretation of all formal concepts (including attributes, predicates and variables, if available) must be provided in order to make clear their intended meaning.
5050	11.4.2.4 Work unit ADV_SPM.1-5
5051 5052	11.4.2.4 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i> it formally proves that the security features enforce the security properties.
5053 5054 5055	11.4.2.4 To determine the enforcement, the evaluator considers the security properties and the security features and verifies that the arguments used in the proof are valid. The proof of correspondence between the security properties and the security features shall be formal.
5056 5057 5058	11.4.2.4 The validity of the security properties shall mean that the TOE is in a secure state. By this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure state.
5059	11.4.2.4 Work unit ADV_SPM.1-6
5060 5061	11.4.2.4 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i> it proves the internal consistency of the TOE security policy model.
5062 5063 5064	11.4.2.4 The proof shall show the absence of contradictions within the TOE security policy model. In determining the absence of contradictions, the evaluator verifies that the arguments used in the proof are valid.
5065 5066 5067 5068 5069 5070	11.4.2.4 Since the TOE security policy model is formal, the proof of its internal consistency shall be formal. It is recognized that a complete formal proof of the internal consistency of the TOE security policy model usually is not possible due to the fundamental nature of formal frameworks. Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE security policy model that prove the internal consistency by means of a combination with generic arguments of the formal framework.
5071 5072	11.4.2.4 ADV_SPM.1.3C <i>The correspondence between the model and the functional specification shall be at the correct level of formality.</i>

- 5073 **11.4.2.4 Work unit ADV_SPM.1-7**
- 5074 **11.4.2.4** The evaluator *shall examine the correspondence between the model and the*
5075 functional specification to determine that a semiformal demonstration of correspondence
5076 between the model and any semiformal functional specification is provided.
- 5077 **11.4.2.4** This work unit is only applicable to a semiformal presentation of the functional
5078 specification, which is required by ADV_FSP.5.2C.
- 5079 **11.4.2.4** A semiformal correspondence is one that results from a structured approach with a
5080 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
5081 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
5082 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 5083 **11.4.2.4** For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
5084 methods’.
- 5085 **11.4.2.4 Work unit ADV_SPM.1-8**
- 5086 **11.4.2.4** The evaluator *shall examine the correspondence between the model and the functional*
5087 specification to determine that a formal proof of correspondence between the model and any
5088 formal functional specification is provided.
- 5089 **11.4.2.4** This work unit is only applicable to a formal presentation of the functional specification,
5090 which is required by ADV_FSP.6.2D.
- 5091 **11.4.2.4** There should be a formal proof of correspondence between the model and any formal
5092 functional specification.
- 5093 **11.4.2.4** The formal proof of correspondence removes all subjective interpretations of its terms
5094 by enlisting well-established mathematical concepts to define the syntax and semantics of the
5095 formal notation and uses rules that support logical reasoning. The security features within the
5096 TOE (which are identified in the formal TSP model) are expressed in a formal specification
5097 language and shown to be satisfied by the formal specification.
- 5098 **11.4.2.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 5099 **11.4.2.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
5100 *specification is consistent and complete with respect to the model.*
- 5101 **11.4.2.4 Work unit ADV_SPM.1-9**
- 5102 **11.4.2.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
5103 TSF interfaces (as articulated in the functional specification) is complete with respect to the
5104 behaviour modelled by the security features.
- 5105 **11.4.2.4** The term “correspondence” here means both the formal proof of correspondence
5106 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
5107 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 5108 **11.4.2.4** In determining completeness of the correspondence, the evaluator considers the
5109 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
5110 features of the TSP model. The demonstration should show that all characteristics belonging to
5111 policies that are required to be modelled have an associated feature description in the TOE
5112 security policy model, and that each feature of the TSP model does occur in the mapping.
- 5113 **11.4.2.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
5114 developer’s side (also confer the application notes above).

5115 **11.4.2.4 Work unit ADV_SPM.1-10**

5116 **11.4.2.4** The evaluator *shall examine* the correspondence to determine that the behaviour at the
5117 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
5118 behaviour modelled by the security features.

5119 **11.4.2.4** The term “correspondence” here means both the formal proof of correspondence
5120 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
5121 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

5122 **11.4.2.4** The meaning of consistency reflects the conventional understanding in contrast to the
5123 internal consistency concept of work unit ADV_SPM.1-6.

5124 **11.4.2.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
5125 security features established in the preceding work unit and verifies that the correspondence
5126 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
5127 behaviour.

5128 **11.4.2.4** For example, if TSFI behaviour dealt with access management on the granularity of
5129 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
5130 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
5131 management for groups of users, then a TSP model describing the security behaviour of the TOE
5132 in terms of individual users would also not be consistent.

5133 **11.4.2.4** As another example, if remote untrusted users had to pass more stringent
5134 authentication procedures than administrators whose only point of access were within a
5135 physically-protected area, then this difference in authentication procedures had to be reflected in
5136 the security features.

5137 **11.4.2.4** TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be
5138 done at a high level to ensure that no large groups of interfaces have been missed (network
5139 protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the
5140 functional specification proceeds.

5141 In making an assessment for this work unit, the evaluator determines that all portions of the TSF
5142 are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF
5143 should have a corresponding interface description, or if there are no corresponding interfaces for a
5144 portion of the TSF, the evaluator determines that that is acceptable.

5145 ISO/IEC 15408-3 ADV_FSP.2.2C: *The functional specification shall describe the purpose and method*
5146 *of use for all TSFI.*

5147 **11.4.2.7.11 Work unit ADV_FSP.2-2**

5148 The evaluator *shall examine* the functional specification to determine that it states the purpose of
5149 each TSFI.

5150 The purpose of a TSFI is a general statement summarising the functionality provided by the
5151 interface. It is not intended to be a complete statement of the actions and results related to the
5152 interface, but rather a statement to help the reader understand in general what the interface is
5153 intended to be used for. The evaluator should not only determine that the purpose exists, but also
5154 that it accurately reflects the TSFI by taking into account other information about the interface,
5155 such as the description of actions and error messages.

5156 **11.4.2.7.12 Work unit ADV_FSP.2-3**

5157 The evaluator *shall examine* the functional specification to determine that the method of use for
5158 each TSFI is given.

5159 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the
 5160 actions and obtain the results associated with the TSFI. The evaluator should be able to determine,
 5161 from reading this material in the functional specification, how to use each interface. This does not
 5162 necessarily mean that there needs to be a separate method of use for each TSFI, as it may be
 5163 possible to describe in general how kernel calls are invoked, for instance, and then identify each
 5164 interface using that general style. Different types of interfaces will require different method of use
 5165 specifications. APIs, network protocol interfaces, system configuration parameters, and hardware
 5166 bus interfaces all have very different methods of use, and this should be taken into account by the
 5167 developer when developing the functional specification, as well as by the evaluator evaluating the
 5168 functional specification.

5169 For administrative interfaces, whose functionality is documented as being inaccessible to
 5170 untrusted users, the evaluator ensures that the method of making the functions inaccessible is
 5171 described in the functional specification. It should be noted that this inaccessibility needs to be
 5172 tested by the developer in their test suite.

5173 The evaluator should not only determine that the set of method of use descriptions exist, but also
 5174 that they accurately cover each TSFI.

5175 ISO/IEC 15408-3 ADV_FSP.2.3C: *The functional specification shall identify and describe all*
 5176 *parameters associated with each TSFI.*

5177 **11.4.2.7.13 Work unit ADV_FSP.2-4**

5178 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies
 5179 all parameters associated with every TSFI.

5180 The evaluator examines the functional specification to ensure that all of the parameters are
 5181 described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the
 5182 behaviour of that interface. For examples, parameters are the arguments supplied to an API; the
 5183 various fields in packet for a given network protocol; the individual key values in the Windows
 5184 Registry; the signals across a set of pins on a chip; etc.

5185 In order to determine that all of the parameters are present in the TSFI, the evaluator should
 5186 examine the rest of the interface description (actions, error messages, etc.) to determine if the
 5187 effects of the parameter are accounted for in the description. The evaluator should also check other
 5188 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 5189 operational user guidance, implementation representation) to see if behaviour or additional
 5190 parameters are described there but not in the functional specification.

5191 **11.4.2.7.14 Work unit ADV_FSP.2-5**

5192 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 5193 accurately describes all parameters associated with every TSFI.

5194 Once all of the parameters have been identified, the evaluator needs to ensure that they are
 5195 accurately described, and that the description of the parameters is complete. A parameter
 5196 description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)*
 5197 could be described as having "parameter i which is an integer"; this is not an acceptable parameter
 5198 description. A description such as "parameter i is an integer that indicates the number of users
 5199 currently logged in to the system" is much more acceptable.

5200 In order to determine that the description of the parameters is complete, the evaluator should
 5201 examine the rest of the interface description (purpose, method of use, actions, error messages, etc.)
 5202 to determine if the descriptions of the parameter(s) are accounted for in the description. The
 5203 evaluator should also check other evidence provided (e.g., TOE design, architectural design,
 5204 operational user guidance, implementation representation) to see if behaviour or additional
 5205 parameters are described there but not in the functional specification.

5206 ISO/IEC 15408-3 ADV_FSP.2.4C: *For each SFR-enforcing TSFI, the functional specification shall*
 5207 *describe the SFR-enforcing actions associated with the TSFI.*

5208 **11.4.2.7.15 Work unit ADV_FSP.2-6**

5209 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 5210 accurately describes the SFR-enforcing actions associated with the SFR-enforcing TSFIs.

5211 If an action available through an interface can be traced to one of the SFRs levied on the TSF, then
 5212 that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but also
 5213 refer to any functionality specified by one of the SFRs contained in the ST. Note that it is possible
 5214 that an interface may have various actions and results, some of which may be SFR-enforcing and
 5215 some of which may not.

5216 The developer is not required to “label” interfaces as SFR-enforcing, and likewise is not required to
 5217 identify actions available through an interface as SFR-enforcing. It is the evaluator's responsibility
 5218 to examine the evidence provided by the developer and determine that the required information is
 5219 present. In the case where the developer has identified the SFR-enforcing TSFI and SFR-enforcing
 5220 actions available through those TSFI, the evaluator must judge completeness and accuracy based
 5221 on other information supplied for the evaluation (e.g., TOE design, security architecture description,
 5222 operational user guidance), and on the other information presented for the interfaces (parameters
 5223 and parameter descriptions, error messages, etc.).

5224 In this case (where the developer has provided only the SFR-enforcing information for SFR-
 5225 enforcing TSFI) the evaluator also ensures that no interfaces have been mis-categorised. This is
 5226 done by examining other information supplied for the evaluation (e.g., TOE design, security
 5227 architecture description, operational user guidance), and the other information presented for the
 5228 interfaces (parameters and parameter descriptions, for example) not labelled as SFR-enforcing.

5229 In the case where the developer has provided the same level of information on all interfaces, the
 5230 evaluator performs the same type of analysis mentioned in the previous paragraphs. The evaluator
 5231 should determine which interfaces are SFR-enforcing and which are not, and subsequently ensure
 5232 that the SFR-enforcing aspects of the SFR-enforcing actions are appropriately described.

5233 The SFR-enforcing actions are those that are visible at any external interface and that provide for
 5234 the enforcement of the SFRs being claimed. For example, if audit requirements are included in the
 5235 ST, then audit-related actions would be SFR-enforcing and therefore must be described, even if the
 5236 result of that action is generally not visible through the invoked interface (as is often the case with
 5237 audit, where a user action at one interface would produce an audit record visible at another
 5238 interface).

5239 The level of description that is required is that sufficient for the reader to understand what role the
 5240 TSFI actions play with respect to the SFR. The evaluator should keep in mind that the description
 5241 should be detailed enough to support the generation (and assessment) of test cases against that
 5242 interface. If the description is unclear or lacking detail such that meaningful testing cannot be
 5243 conducted against the TSFI, it is likely that the description is inadequate.

5244 ISO/IEC 15408-3 ADV_FSP.2.5C: *For each SFR-enforcing TSFI, the functional specification shall*
 5245 *describe direct error messages resulting from processing associated with the SFR-enforcing actions.*

5246 **11.4.2.7.16 Work unit ADV_FSP.2-7**

5247 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 5248 accurately describes error messages that may result from SFR-enforcing actions associated with
 5249 each SFR-enforcing TSFI.

5250 This work unit should be performed in conjunction with, or after, work unit ADV_FSP.2-6 in order
 5251 to ensure the set of SFR-enforcing TSFI and SFR-enforcing actions is correctly identified. The

5252 developer may provide more information than is required (for example, all error messages
5253 associated with each interface), in which the case the evaluator should restrict their assessment of
5254 completeness and accuracy to only those that they determine to be associated with SFR-enforcing
5255 actions of SFR-enforcing TSFI.

5256 Errors can take many forms, depending on the interface being described. For an API, the interface
5257 itself may return an error code, set a global error condition, or set a certain parameter with an
5258 error code. For a configuration file, an incorrectly configured parameter may cause an error
5259 message to be written to a log file. For a hardware PCI card, an error condition may raise a signal
5260 on the bus, or trigger an exception condition to the CPU.

5261 **11.4.2.8 Errors (and the associated error messages) come about through the invocation of an interface. The**
5262 **processing that occurs in response to the interface invocation may encounter error conditions,**
5263 **which trigger (through an implementation-specific mechanism) an error message to be generated.**
5264 **In some instances, this may be a return value from the interface itself; in other instances a global**
5265 **value may be set and checked after the invocation of an interface. It is likely that a TOE will have a**
5266 **number of low-level error messages that may result from fundamental resource conditions, such as**
5267 **“disk full” or “resource locked”. While these error messages may map to a large number of TSFI,**
5268 **they could be used to detect instances where detail from an interface description has been omitted.**
5269 **For instance, a TSFI that produces a “disk full” message, but has no obvious description of why that**
5270 **TSFI should cause an access to the disk in its description of actions, might cause the evaluator to**
5271 **examine other evidence (Security Architecture (ADV_ARC), Objectives**

5272 **11.4.2.8** The objectives of this sub-activity are to determine whether the formal security policy model of
5273 the TSF clearly and consistently describes the rules and characteristics of the security policies
5274 and whether this description corresponds with the description of security functions in the
5275 functional specification.

5276 **11.4.2.8 Input**

5277 **11.4.2.8** The evaluation evidence for this sub-activity is:

5278 **11.4.2.8** the ST;

5279 **11.4.2.8** the functional specification;

5280 **11.4.2.8** formal security policy model (ADV_SPM.1.1D);

5281 **11.4.2.8** formal proof of correspondence between the model and any formal functional specification
5282 (ADV_SPM.1.3D);

5283 **11.4.2.8** demonstration of correspondence between the model and the functional specification
5284 (ADV_SPM.1.4D).

5285 **11.4.2.8 Application notes**

5286 **11.4.2.8** This activity applies to cases where the developer has provided a formal security policy model of
5287 the TOE.

5288 **11.4.2.8** A formal TOE security policy model is a representation of the rules (synonymously termed
5289 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
5290 Their formal counterparts are called security properties and security features, respectively. The
5291 representation includes but is not limited to algebraic specifications, finite state machines and
5292 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
5293 model is accompanied by an informal interpretation explaining how the rules and characteristics
5294 are mapped to the respective properties and features.

5295 **11.4.2.8** The creation of a formal security policy model helps to identify and eliminate
 5296 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
 5297 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
 5298 judgement of how well the developer has understood the security functionality being
 5299 implemented and whether there are inconsistencies between the security requirements and the
 5300 TOE design. The confidence in the model is accompanied by a proof that it contains no
 5301 inconsistencies.

5302 **11.4.2.8** A formal security model is a precise formal presentation of the important aspects of
 5303 security and their relationship to the behaviour of the TOE; it identifies the set of rules
 5304 (principles) that defines the TOE security policy and the set of practises (characteristics) that
 5305 regulates how the TSF manages, protects, and otherwise controls the system resources. The
 5306 model includes the set of restrictions and properties that specify how information and computing
 5307 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
 5308 engineering arguments showing that these restrictions and properties play a key role in the
 5309 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
 5310 as well as ancillary text to explain the model and to provide it with context. The security
 5311 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
 5312 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

5313 **11.4.2.8** The Security Policy Model of the TOE is informally abstracted from its realisation by
 5314 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 5315 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 5316 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 5317 are always prone to fallacies; especially if relationships among subjects, objects and operations
 5318 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 5319 characteristics of the security policy model are mapped to respective properties and features
 5320 within some formal system, whose rigour and strength can afterwards be used to obtain the
 5321 security properties by means of theorems and formal proof.

5322 **11.4.2.8** While the term "formal security policy model" is used in academic circles, the CC's
 5323 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
 5324 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 5325 of SFRs being claimed.

5326 **11.4.2.8** The term security policy has traditionally been associated with only access control
 5327 policies, whether label-based (mandatory access control) or user-based (discretionary access
 5328 control). However, a security policy is not limited to access control; there are also audit policies,
 5329 identification policies, authentication policies, encryption policies, management policies, and any
 5330 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 5331 contains an assignment for identifying these policies that are formally modelled.

5332 **11.4.2.8** It is recognized that not all policies can be formally modelled for all TOEs. This is
 5333 because either a given policy can not be formally modelled in the otherwise well suited
 5334 framework, or because the nature of the TOE renders impossible the modelling of policies that
 5335 would otherwise be possible to model.

5336 **11.4.2.8 Action ADV_SPM.1.1E**

5337 **11.4.2.8 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 5338 *text as required, and identify the security policies of the TSF that are modelled.*

5339 **11.4.2.8 Work unit ADV_SPM.1-1**

5340 **11.4.2.8** The evaluator *shall examine* the TOE security policy model to determine that it is
 5341 written in a formal style.

- 5342 **11.4.2.8** The evaluator identifies the formal framework upon which the TOE security policy
 5343 model is based and ensures that it is founded on well established mathematical concepts. **They**
 5344 also identify **the security properties and features addressed in the application notes and**
 5345 ensure the formalization of at least one security policy.
- 5346 **11.4.2.8** For guidance on formal methods refer to ISO/IEC **15408-3**
- 5347 **11.4.2.8 Work unit ADV_SPM.1-2**
- 5348 **11.4.2.8** The evaluator *shall examine the TOE security policy model to determine that it*
 5349 contains all necessary informal explanatory text.
- 5350 **11.4.2.8** Supporting narrative descriptions are necessary for all parts of the model (for example,
 5351 to make clear the meaning of any formal notation and how they are used) including the security
 5352 properties and features.
- 5353 **11.4.2.8 Work unit ADV_SPM.1-3**
- 5354 **11.4.2.8** The evaluator *shall examine the TOE security policy model to determine that all*
 5355 security policies of the TSF are identified that are modelled.
- 5356 **11.4.2.8** The evaluator determines whether the SPM identifies the security policies for which a
 5357 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 5358 of the modelled policies.
- 5359 **11.4.2.8** The evaluator determines whether the list of security policies identified by the SPM is
 5360 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 5361 **11.4.2.8** The evaluator determines whether for each security policy identified by the SPM a
 5362 model is in fact provided.
- 5363 **11.4.2.8 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 5364 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 5365 *not secure.*
- 5366 **11.4.2.8 Work unit ADV_SPM.1-4**
- 5367 **11.4.2.8** The evaluator *shall examine the principles and characteristics of the security policies*
 5368 to determine that the modelled security behaviour of the TOE is clearly articulated.
- 5369 **11.4.2.8** The security policies are expressed in terms of security principles (rules) which are
 5370 modelled by security properties and define the secure state of the TOE. For example, a model
 5371 based on state transitions could describe the security policies in terms of principles of its states,
 5372 identify its initial state, and define what it means to be a secure state.
- 5373 **11.4.2.8** The evaluator determines that the security policies are reflected within their formal
 5374 counterparts of the TSP model.
- 5375 **11.4.2.8** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 5376 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 5377 resources including attributes and conditions of the TOE) which are modelled by security
 5378 features. For example, a model based on state transitions could describe the characteristics as
 5379 possible actions in each secure state in a level of detail sufficient to decide into which state the
 5380 TOE will be transformed by that action.
- 5381 **11.4.2.8** Together the security principles and characteristics describe the entire security posture
 5382 of the TOE.

5383	11.4.2.8	In the context of a formal TOE security policy model the security behaviour is
5384		considered to be clearly articulated only if an adequate mapping from principles and
5385		characteristics to their respective formal counterparts properties and features has been given.
5386		The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
5387		detailed enough to allow for correct identification of all security objectives and the relation to the
5388		security environment.
5389	11.4.2.8	The above condition for clear articulation is necessary but not sufficient. An informal
5390		interpretation of all formal concepts (including attributes, predicates and variables, if available)
5391		must be provided in order to make clear their intended meaning.
5392	11.4.2.8 Work unit ADV_SPM.1-5	
5393	11.4.2.8	The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
5394		it formally proves that the security features enforce the security properties.
5395	11.4.2.8	To determine the enforcement, the evaluator considers the security properties and the
5396		security features and verifies that the arguments used in the proof are valid. The proof of
5397		correspondence between the security properties and the security features shall be formal.
5398	11.4.2.8	The validity of the security properties shall mean that the TOE is in a secure state. By
5399		this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
5400		state.
5401	11.4.2.8 Work unit ADV_SPM.1-6	
5402	11.4.2.8	The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
5403		it proves the internal consistency of the TOE security policy model.
5404	11.4.2.8	The proof shall show the absence of contradictions within the TOE security policy
5405		model. In determining the absence of contradictions, the evaluator verifies that the arguments
5406		used in the proof are valid.
5407	11.4.2.8	Since the TOE security policy model is formal, the proof of its internal consistency shall
5408		be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
5409		security policy model usually is not possible due to the fundamental nature of formal frameworks.
5410		Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
5411		security policy model that prove the internal consistency by means of a combination with generic
5412		arguments of the formal framework.
5413	11.4.2.8 ADV_SPM.1.3C	<i>The correspondence between the model and the functional</i>
5414		<i>specification shall be at the correct level of formality.</i>
5415	11.4.2.8 Work unit ADV_SPM.1-7	
5416	11.4.2.8	The evaluator <i>shall examine the correspondence between the model and the functional</i>
5417		specification to determine that a semiformal demonstration of correspondence between the
5418		model and any semiformal functional specification is provided.
5419	11.4.2.8	This work unit is only applicable to a semiformal presentation of the functional
5420		specification, which is required by ADV_FSP.5.2C.
5421	11.4.2.8	A semiformal correspondence is one that results from a structured approach with a
5422		substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
5423		mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
5424		terms, and so it provides less ambiguity than would exist in an informal correspondence.

- 5425 **11.4.2.8** For guidance on semiformal methods refer to Annex 3.1.1 ‘**Semiformal and formal**
5426 **methods**’.
- 5427 **11.4.2.8 Work unit ADV_SPM.1-8**
- 5428 **11.4.2.8** The evaluator *shall examine the correspondence between the model and the functional*
5429 *specification to determine that a formal proof of correspondence between the model and any*
5430 *formal functional specification is provided.*
- 5431 **11.4.2.8** This work unit is only applicable to a formal presentation of the functional specification,
5432 which is required by ADV_FSP.6.2D.
- 5433 **11.4.2.8** There should be a formal proof of correspondence between the model and any formal
5434 functional specification.
- 5435 **11.4.2.8** The formal proof of correspondence removes all subjective interpretations of its terms
5436 by enlisting well-established mathematical concepts to define the syntax and semantics of the
5437 formal notation and uses rules that support logical reasoning. The security features within the
5438 TOE (which are identified in the formal TSP model) are expressed in a formal specification
5439 language and shown to be satisfied by the formal specification.
- 5440 **11.4.2.8** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 5441 **11.4.2.8 ADV_SPM.1.4C** *The correspondence shall show that the functional*
5442 *specification is consistent and complete with respect to the model.*
- 5443 **11.4.2.8 Work unit ADV_SPM.1-9**
- 5444 **11.4.2.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
5445 *TSF interfaces (as articulated in the functional specification) is complete with respect to the*
5446 *behaviour modelled by the security features.*
- 5447 **11.4.2.8** The term “correspondence” here means both the formal proof of correspondence
5448 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
5449 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 5450 **11.4.2.8** In determining completeness of the correspondence, the evaluator considers the
5451 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
5452 features of the TSP model. The demonstration should show that all characteristics belonging to
5453 policies that are required to be modelled have an associated feature description in the TOE
5454 security policy model, and that each feature of the TSP model does occur in the mapping.
- 5455 **11.4.2.8** Abstention from formally modelling TSFI behaviour always calls for justification on the
5456 developer’s side (also confer the application notes above).
- 5457 **11.4.2.8 Work unit ADV_SPM.1-10**
- 5458 **11.4.2.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
5459 *TSF interfaces (as articulated in the functional specification) is consistent with respect to the*
5460 *behaviour modelled by the security features.*
- 5461 **11.4.2.8** The term “correspondence” here means both the formal proof of correspondence
5462 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
5463 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 5464 **11.4.2.8** The meaning of consistency reflects the conventional understanding in contrast to the
5465 internal consistency concept of work unit ADV_SPM.1-6.

- 5466 **11.4.2.8** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 5467 security features established in the preceding work unit and verifies that the correspondence
 5468 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 5469 behaviour.
- 5470 **11.4.2.8** For example, if TSFI behaviour dealt with access management on the granularity of
 5471 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 5472 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 5473 management for groups of users, then a TSP model describing the security behaviour of the TOE
 5474 in terms of individual users would also not be consistent.
- 5475 **11.4.2.8** As another example, if remote untrusted users had to pass more stringent
 5476 authentication procedures than administrators whose only point of access were within a
 5477 physically-protected area, then this difference in authentication procedures had to be reflected in
 5478 the security features.
- 5479 **11.4.2.8** TOE design (ADV_TDS)) related that TSFI to determine if the description is accurate.
- 5480 In order to determine that the description of the error messages of a TSFI is accurate and complete,
 5481 the evaluator measures the interface description against the other evidence provided for the
 5482 evaluation (e.g., TOE design, security architecture description, operational user guidance), as well
 5483 as other evidence available for that TSFI (parameters, analysis from work unit ADV_FSP.2-6).
- 5484 ISO/IEC 15408-3 ADV_FSP.2.6C: *The tracing shall demonstrate that the SFRs trace to TSFIs in the*
 5485 *functional specification.*
- 5486 **11.4.2.11.11 Work unit ADV_FSP.2-8**
- 5487 The evaluator **shall check** that the tracing links the SFRs to the corresponding TSFIs.
- 5488 The tracing is provided by the developer to serve as a guide to which SFRs are related to which
 5489 TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the
 5490 following work units, in which the evaluator verifies its completeness and accuracy.
- 5491 **11.4.2.12 Action ADV_FSP.2.2E**
- 5492 **11.4.2.12.1 Work unit ADV_FSP.2-9**
- 5493 The evaluator **shall examine** the functional specification to determine that it is a complete
 5494 instantiation of the SFRs.
- 5495 To ensure that all SFRs are covered by the functional specification, as well as the test coverage
 5496 analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.2-8 a map between
 5497 the TOE security functional requirements and the TSFI. Note that this map may have to be at a level
 5498 of detail below the component or even element level of the requirements, because of operations
 5499 (assignments, refinements, selections) performed on the functional requirement by the ST author.
- 5500 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained,
 5501 for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three
 5502 different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and
 5503 claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to
 5504 TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper
 5505 interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of
 5506 parameters for a given interface.
- 5507 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 5508 boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the
 5509 TSFI. The analysis for those requirements will be performed in the analysis for the TOE design

5510 (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions,
5511 and error messages associated with TSFIs must be fully specified, the evaluator should be able to
5512 determine if all aspects of an SFR appear to be implemented at the interface level.

5513 **11.4.2.12.2 Work unit ADV_FSP.2-10**

5514 The evaluator *shall examine* the functional specification to determine that it is an accurate
5515 instantiation of the SFRs.

5516 For each functional requirement in the ST that results in effects visible at the TSF boundary, the
5517 information in the associated TSFI for that requirement specifies the required functionality
5518 described by the requirement. For example, if the ST contains a requirement for access control lists,
5519 and the only TSFI that map to that requirement specify functionality for Unix-style protection bits,
5520 then the functional specification is not accurate with respect to the requirements.

5521 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
5522 boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements
5523 to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE
5524 design (ADV_TDS) when included in the ST.

5525 **11.4.3 Evaluation of sub-activity (ADV_FSP.3)**

5526 **11.4.3.1 Objectives**

5527 The objective of this sub-activity is to determine whether the developer has provided a description
5528 of the TSFIs in terms of their purpose, method of use, and parameters. In addition, the actions,
5529 results and error messages of each TSFI are also described sufficiently that it can be determined
5530 whether they are SFR-enforcing, with the SFR-enforcing TSFI being described in more detail than
5531 other TSFIs.

5532 **11.4.3.2 Input**

5533 The evaluation evidence for this sub-activity that is required by the work-units is:

- 5534 a) the ST;
5535 b) the functional specification;
5536 c) the TOE design.

5537 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

- 5538 a) the security architecture description;
5539 b) the implementation representation;
5540 c) the TSF internals description;
5541 d) the operational user guidance;

5542 **11.4.3.3 Action ADV_FSP.3.1E**

5543 ISO/IEC 15408-3 ADV_FSP.3.1C: *The functional specification shall completely represent the TSF.*

5544 **11.4.3.3.1 Work unit ADV_FSP.3-1**

5545 The evaluator *shall examine* the functional specification to determine that the TSF is fully
5546 represented.

5547 **11.4.3.4 The identification of the TSFI is a necessary prerequisite to all other activities in**
 5548 **this sub-activity. The TSF must be identified (done as part of the Objectives)**

5549 **11.4.3.4** The objectives of this sub-activity are to determine whether the formal security policy model of
 5550 the TSF clearly and consistently describes the rules and characteristics of the security policies
 5551 and whether this description corresponds with the description of security functions in the
 5552 functional specification.

5553 **11.4.3.4 Input**

5554 **11.4.3.4** The evaluation evidence for this sub-activity is:

5555 **11.4.3.4** the ST;

5556 **11.4.3.4** the functional specification;

5557 **11.4.3.4** formal security policy model (ADV_SPM.1.1D);

5558 **11.4.3.4** formal proof of correspondence between the model and any formal functional specification
 5559 (ADV_SPM.1.3D);

5560 **11.4.3.4** demonstration of correspondence between the model and the functional specification
 5561 (ADV_SPM.1.4D).

5562 **11.4.3.4 Application notes**

5563 **11.4.3.4** This activity applies to cases where the developer has provided a formal security policy model of
 5564 the TOE.

5565 **11.4.3.4** A formal TOE security policy model is a representation of the rules (synonymously termed
 5566 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
 5567 Their formal counterparts are called security properties and security features, respectively. The
 5568 representation includes but is not limited to algebraic specifications, finite state machines and
 5569 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
 5570 model is accompanied by an informal interpretation explaining how the rules and characteristics
 5571 are mapped to the respective properties and features.

5572 **11.4.3.4** The creation of a formal security policy model helps to identify and eliminate ambiguous,
 5573 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
 5574 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
 5575 of how well the developer has understood the security functionality being implemented and
 5576 whether there are inconsistencies between the security requirements and the TOE design. The
 5577 confidence in the model is accompanied by a proof that it contains no inconsistencies.

5578 **11.4.3.4** A formal security model is a precise formal presentation of the important aspects of security and
 5579 their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that
 5580 defines the TOE security policy and the set of practises (characteristics) that regulates how the
 5581 TSF manages, protects, and otherwise controls the system resources. The model includes the set
 5582 of restrictions and properties that specify how information and computing resources are
 5583 prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
 5584 arguments showing that these restrictions and properties play a key role in the enforcement of
 5585 the SFRs. It consists both of the formalisms that express the security functionality, as well as
 5586 ancillary text to explain the model and to provide it with context. The security behaviour of the
 5587 TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
 5588 the TOE and with its operational environment), as well as its internal behaviour.

5589 **11.4.3.4** The Security Policy Model of the TOE is informally abstracted from its realisation by considering
 5590 the proposed security requirements of the ST. The informal abstraction is taken to be successful if

5591 the TOE's principles turn out to be enforced by its characteristics. The purpose of formal methods
5592 lies within the enhancement of the rigour of enforcement. Informal arguments are always prone
5593 to fallacies; especially if relationships among subjects, objects and operations get more and more
5594 involved. In order to minimise the risk of insecure state arrivals the rules and characteristics of
5595 the security policy model are mapped to respective properties and features within some formal
5596 system, whose rigour and strength can afterwards be used to obtain the security properties by
5597 means of theorems and formal proof.

5598 **11.4.3.4** While the term “formal security policy model” is used in academic circles, the CC's
5599 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
5600 claimed. Therefore, the formal security policy model is merely a formal representation of the set
5601 of SFRs being claimed.

5602 **11.4.3.4** The term security policy has traditionally been associated with only access control
5603 policies, whether label-based (mandatory access control) or user-based (discretionary access
5604 control). However, a security policy is not limited to access control; there are also audit policies,
5605 identification policies, authentication policies, encryption policies, management policies, and any
5606 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
5607 contains an assignment for identifying these policies that are formally modelled.

5608 **11.4.3.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
5609 because either a given policy can not be formally modelled in the otherwise well suited
5610 framework, or because the nature of the TOE renders impossible the modelling of policies that
5611 would otherwise be possible to model.

5612 **11.4.3.4 Action ADV_SPM.1.1E**

5613 **11.4.3.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
5614 *text as required, and identify the security policies of the TSF that are modelled.*

5615 **11.4.3.4 Work unit ADV_SPM.1-1**

5616 **11.4.3.4** The evaluator ***shall examine the TOE security policy model to determine that it is***
5617 ***written in a formal style.***

5618 **11.4.3.4** The evaluator identifies the formal framework upon which the TOE security policy
5619 model is based and ensures that it is founded on well established mathematical concepts. **They**
5620 **also identify the security properties and features addressed in the application notes and ensure**
5621 **the formalization of at least one security policy.**

5622 **11.4.3.4** For guidance on formal methods refer to ISO/IEC 15408-3

5623 **11.4.3.4 Work unit ADV_SPM.1-2**

5624 **11.4.3.4** The evaluator ***shall examine the TOE security policy model to determine that it***
5625 ***contains all necessary informal explanatory text.***

5626 **11.4.3.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
5627 to make clear the meaning of any formal notation and how they are used) including the security
5628 properties and features.

5629 **11.4.3.4 Work unit ADV_SPM.1-3**

5630 **11.4.3.4** The evaluator ***shall examine the TOE security policy model to determine that all***
5631 ***security policies of the TSF are identified that are modelled.***

- 5632 **11.4.3.4** The evaluator determines whether the SPM identifies the security policies for which a
 5633 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 5634 of the modelled policies.
- 5635 **11.4.3.4** The evaluator determines whether the list of security policies identified by the SPM is
 5636 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 5637 **11.4.3.4** The evaluator determines whether for each security policy identified by the SPM a
 5638 model is in fact provided.
- 5639 **11.4.3.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 5640 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 5641 *not secure.*
- 5642 **11.4.3.4 Work unit ADV_SPM.1-4**
- 5643 **11.4.3.4** The evaluator *shall examine the principles and characteristics of the security policies*
 5644 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 5645 **11.4.3.4** The security policies are expressed in terms of security principles (rules) which are
 5646 modelled by security properties and define the secure state of the TOE. For example, a model
 5647 based on state transitions could describe the security policies in terms of principles of its states,
 5648 identify its initial state, and define what it means to be a secure state.
- 5649 **11.4.3.4** The evaluator determines that the security policies are reflected within their formal
 5650 counterparts of the TSP model.
- 5651 **11.4.3.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 5652 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 5653 resources including attributes and conditions of the TOE) which are modelled by security
 5654 features. For example, a model based on state transitions could describe the characteristics as
 5655 possible actions in each secure state in a level of detail sufficient to decide into which state the
 5656 TOE will be transformed by that action.
- 5657 **11.4.3.4** Together the security principles and characteristics describe the entire security posture
 5658 of the TOE.
- 5659 **11.4.3.4** In the context of a formal TOE security policy model the security behaviour is
 5660 considered to be clearly articulated only if an adequate mapping from principles and
 5661 characteristics to their respective formal counterparts properties and features has been given.
 5662 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 5663 detailed enough to allow for correct identification of all security objectives and the relation to the
 5664 security environment.
- 5665 **11.4.3.4** The above condition for clear articulation is necessary but not sufficient. An informal
 5666 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 5667 must be provided in order to make clear their intended meaning.
- 5668 **11.4.3.4 Work unit ADV_SPM.1-5**
- 5669 **11.4.3.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
 5670 *it formally proves that the security features enforce the security properties.*
- 5671 **11.4.3.4** To determine the enforcement, the evaluator considers the security properties and the
 5672 security features and verifies that the arguments used in the proof are valid. The proof of
 5673 correspondence between the security properties and the security features shall be formal.

- 5674 **11.4.3.4** The validity of the security properties shall mean that the TOE is in a secure state. By
 5675 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 5676 state.
- 5677 **11.4.3.4 Work unit ADV_SPM.1-6**
- 5678 **11.4.3.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
 5679 it proves the internal consistency of the TOE security policy model.
- 5680 **11.4.3.4** The proof shall show the absence of contradictions within the TOE security policy
 5681 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 5682 used in the proof are valid.
- 5683 **11.4.3.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
 5684 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 5685 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 5686 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 5687 security policy model that prove the internal consistency by means of a combination with generic
 5688 arguments of the formal framework.
- 5689 **11.4.3.4 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 5690 *specification shall be at the correct level of formality.*
- 5691 **11.4.3.4 Work unit ADV_SPM.1-7**
- 5692 **11.4.3.4** The evaluator *shall examine the correspondence between the model and the functional*
 5693 *specification to determine that a semiformal demonstration of correspondence between the*
 5694 *model and any semiformal functional specification is provided.*
- 5695 **11.4.3.4** This work unit is only applicable to a semiformal presentation of the functional
 5696 specification, which is required by ADV_FSP.5.2C.
- 5697 **11.4.3.4** A semiformal correspondence is one that results from a structured approach with a
 5698 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 5699 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 5700 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 5701 **11.4.3.4** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 5702 **methods**'.
- 5703 **11.4.3.4 Work unit ADV_SPM.1-8**
- 5704 **11.4.3.4** The evaluator *shall examine the correspondence between the model and the functional*
 5705 *specification to determine that a formal proof of correspondence between the model and any*
 5706 *formal functional specification is provided.*
- 5707 **11.4.3.4** This work unit is only applicable to a formal presentation of the functional specification,
 5708 which is required by ADV_FSP.6.2D.
- 5709 **11.4.3.4** There should be a formal proof of correspondence between the model and any formal
 5710 functional specification.
- 5711 **11.4.3.4** The formal proof of correspondence removes all subjective interpretations of its terms
 5712 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 5713 formal notation and uses rules that support logical reasoning. The security features within the
 5714 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 5715 language and shown to be satisfied by the formal specification.

- 5716 **11.4.3.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 5717 **11.4.3.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 5718 *specification is consistent and complete with respect to the model.*
- 5719 **11.4.3.4 Work unit ADV_SPM.1-9**
- 5720 **11.4.3.4** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 5721 ***TSF interfaces (as articulated in the functional specification) is complete with respect to the***
 5722 ***behaviour modelled by the security features.***
- 5723 **11.4.3.4** The term “correspondence” here means both the formal proof of correspondence
 5724 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 5725 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 5726 **11.4.3.4** In determining completeness of the correspondence, the evaluator considers the
 5727 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 5728 features of the TSP model. The demonstration should show that all characteristics belonging to
 5729 policies that are required to be modelled have an associated feature description in the TOE
 5730 security policy model, and that each feature of the TSP model does occur in the mapping.
- 5731 **11.4.3.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
 5732 developer’s side (also confer the application notes above).
- 5733 **11.4.3.4 Work unit ADV_SPM.1-10**
- 5734 **11.4.3.4** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 5735 ***TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
 5736 ***behaviour modelled by the security features.***
- 5737 **11.4.3.4** The term “correspondence” here means both the formal proof of correspondence
 5738 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 5739 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 5740 **11.4.3.4** The meaning of consistency reflects the conventional understanding in contrast to the
 5741 internal consistency concept of work unit ADV_SPM.1-6.
- 5742 **11.4.3.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 5743 security features established in the preceding work unit and verifies that the correspondence
 5744 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 5745 behaviour.
- 5746 **11.4.3.4** For example, if TSFI behaviour dealt with access management on the granularity of
 5747 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 5748 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 5749 management for groups of users, then a TSP model describing the security behaviour of the TOE
 5750 in terms of individual users would also not be consistent.
- 5751 **11.4.3.4** As another example, if remote untrusted users had to pass more stringent
 5752 authentication procedures than administrators whose only point of access were within a
 5753 physically-protected area, then this difference in authentication procedures had to be reflected in
 5754 the security features.
- 5755 **11.4.3.4** TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be
 5756 done at a high level to ensure that no large groups of interfaces have been missed (network
 5757 protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the
 5758 functional specification proceeds.

5759 In making an assessment for this work unit, the evaluator determines that all portions of the TSF
5760 are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF
5761 should have a corresponding interface description, or if there are no corresponding interfaces for a
5762 portion of the TSF, the evaluator determines that that is acceptable.

5763 ISO/IEC 15408-3 ADV_FSP.3.2C: *The functional specification shall describe the purpose and method*
5764 *of use for all TSFI.*

5765 **11.4.3.7.11 Work unit ADV_FSP.3-2**

5766 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of
5767 each TSFI.

5768 The purpose of a TSFI is a general statement summarising the functionality provided by the
5769 interface. It is not intended to be a complete statement of the actions and results related to the
5770 interface, but rather a statement to help the reader understand in general what the interface is
5771 intended to be used for. The evaluator should not only determine that the purpose exists, but also
5772 that it accurately reflects the TSFI by taking into account other information about the interface,
5773 such as the description of actions and error messages.

5774 **11.4.3.7.12 Work unit ADV_FSP.3-3**

5775 The evaluator ***shall examine*** the functional specification to determine that the method of use for
5776 each TSFI is given.

5777 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the
5778 actions and obtain the results associated with the TSFI. The evaluator should be able to determine,
5779 from reading this material in the functional specification, how to use each interface. This does not
5780 necessarily mean that there needs to be a separate method of use for each TSFI, as it may be
5781 possible to describe in general how kernel calls are invoked, for instance, and then identify each
5782 interface using that general style. Different types of interfaces will require different method of use
5783 specifications. APIs, network protocol interfaces, system configuration parameters, and hardware
5784 bus interfaces all have very different methods of use, and this should be taken into account by the
5785 developer when developing the functional specification, as well as by the evaluator evaluating the
5786 functional specification.

5787 For administrative interfaces whose functionality is documented as being inaccessible to untrusted
5788 users, the evaluator ensures that the method of making the functions inaccessible is described in
5789 the functional specification. It should be noted that this inaccessibility needs to be tested by the
5790 developer in their test suite.

5791 The evaluator should not only determine that the set of method of use descriptions exist, but also
5792 that they accurately cover each TSFI.

5793 ISO/IEC 15408-3 ADV_FSP.3.3C: *The functional specification shall identify and describe all*
5794 *parameters associated with each TSFI.*

5795 **11.4.3.7.13 Work unit ADV_FSP.3-4**

5796 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies
5797 all parameters associated with every TSFI.

5798 The evaluator examines the functional specification to ensure that all of the parameters are
5799 described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the
5800 behaviour of that interface. For examples, parameters are the arguments supplied to an API; the
5801 various fields in packet for a given network protocol; the individual key values in the Windows
5802 Registry; the signals across a set of pins on a chip; etc.

5803 In order to determine that all of the parameters are present in the TSFI, the evaluator should
 5804 examine the rest of the interface description (actions, error messages, etc.) to determine if the
 5805 effects of the parameter are accounted for in the description. The evaluator should also check other
 5806 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 5807 operational user guidance, implementation representation) to see if behaviour or additional
 5808 parameters are described there but not in the functional specification.

5809 **11.4.3.7.14 Work unit ADV_FSP.3-5**

5810 The evaluator *shall examine* the presentation of the TSFI to determine that it completely and
 5811 accurately describes all parameters associated with every TSFI.

5812 Once all of the parameters have been identified, the evaluator needs to ensure that they are
 5813 accurately described, and that the description of the parameters is complete. A parameter
 5814 description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)*
 5815 could be described as having “parameter i which is an integer”; this is not an acceptable parameter
 5816 description. A description such as “parameter i is an integer that indicates the number of users
 5817 currently logged in to the system” is much more acceptable.

5818 In order to determine that the description of the parameters is complete, the evaluator should
 5819 examine the rest of the interface description (purpose, method of use, actions, error messages, etc.)
 5820 to determine if the descriptions of the parameter(s) are accounted for in the description. The
 5821 evaluator should also check other evidence provided (e.g., TOE design, architectural design,
 5822 operational user guidance, implementation representation) to see if behaviour or additional
 5823 parameters are described there but not in the functional specification.

5824 ISO/IEC 15408-3 ADV_FSP.3.4C: *For each SFR-enforcing TSFI, the functional specification shall*
 5825 *describe the SFR-enforcing actions associated with the TSFI.*

5826 **11.4.3.7.15 Work unit ADV_FSP.3-6**

5827 The evaluator *shall examine* the presentation of the TSFI to determine that it completely and
 5828 accurately describes the SFR-enforcing actions associated with the SFR-enforcing TSFIs.

5829 If an action available through an interface plays a role in enforcing any security policy on the TOE
 5830 (that is, if one of the actions of the interface can be traced to one of the SFRs levied on the TSF),
 5831 then that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but
 5832 also refer to any functionality specified by one of the SFRs contained in the ST. Note that it is
 5833 possible that an interface may have various actions and results, some of which may be SFR-
 5834 enforcing and some of which may not.

5835 The developer is not required to “label” interfaces as SFR-enforcing, and likewise is not required to
 5836 identify actions available through an interface as SFR-enforcing. It is the evaluator's responsibility
 5837 to examine the evidence provided by the developer and determine that the required information is
 5838 present. In the case where the developer has identified the SFR-enforcing TSFI and SFR-enforcing
 5839 actions available through those TSFI, the evaluator must judge completeness and accuracy based
 5840 on other information supplied for the evaluation (e.g., TOE design, security architecture description,
 5841 operational user guidance), and on the other information presented for the interfaces (parameters
 5842 and parameter descriptions, error messages, etc.).

5843 In this case (developer has provided only the SFR-enforcing information for SFR-enforcing TSFI)
 5844 the evaluator also ensures that no interfaces have been mis-categorised. This is done by examining
 5845 other information supplied for the evaluation (e.g., TOE design, security architecture description,
 5846 operational user guidance), and the other information presented for the interfaces (parameters
 5847 and parameter descriptions, for example) not labelled as SFR-enforcing. The analysis done for
 5848 work units ADV_FSP.3-7 and ADV_FSP.3-8 are also used in making this determination.

In the case where the developer has provided the same level of information on all interfaces, the evaluator performs the same type of analysis mentioned in the previous paragraphs. The evaluator should determine which interfaces are SFR-enforcing and which are not, and subsequently ensure that the SFR-enforcing aspects of the SFR-enforcing actions are appropriately described. Note that in this case, the evaluator should be able to perform the bulk of the work associated with work unit ADV_FSP.3-8 in the course of performing this SFR-enforcing analysis.

The SFR-enforcing actions are those that are visible at any external interface and that provide for the enforcement of the SFRs being claimed. For example, if audit requirements are included in the ST, then audit-related actions would be SFR-enforcing and therefore must be described, even if the result of that action is generally not visible through the invoked interface (as is often the case with audit, where a user action at one interface would produce an audit record visible at another interface).

The level of description that is required is that sufficient for the reader to understand what role the TSFI actions play with respect to the SFR. The evaluator should keep in mind that the description should be detailed enough to support the generation (and assessment) of test cases against that interface. If the description is unclear or lacking detail such that meaningful testing cannot be conducted against the TSFI, it is likely that the description is inadequate.

ISO/IEC 15408-3 ADV_FSP.3.5C: *For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from SFR-enforcing actions and exceptions associated with invocation of the TSFI.*

11.4.3.7.16 Work unit ADV_FSP.3-7

The evaluator **shall examine** the presentation of the TSFI to determine that it completely and accurately describes error messages that may result from an invocation of each SFR-enforcing TSFI.

This work unit should be performed in conjunction with, or after, work unit ADV_FSP.3-6 in order to ensure the set of SFR-enforcing TSFI is correctly identified. The evaluator should note that the requirement and associated work unit is that all direct error messages associated with an SFR-enforcing TSFI must be described, that are associated with SFR-enforcing actions. This is because at this level of assurance, the “extra” information provided by the error message descriptions should be used in determining whether all of the SFR-enforcing aspects of an interface have been appropriately described. For instance, if an error message associated with a TSFI (e.g., “access denied”) indicated that an SFR-enforcing decision or action had taken place, but in the description of the SFR-enforcing actions there was no mention of that particular SFR-enforcing mechanism, then the description may not be complete.

Errors can take many forms, depending on the interface being described. For an API, the interface itself may return an error code, set a global error condition, or set a certain parameter with an error code. For a configuration file, an incorrectly configured parameter may cause an error message to be written to a log file. For a hardware PCI card, an error condition may raise a signal on the bus, or trigger an exception condition to the CPU.

11.4.3.8 Errors (and the associated error messages) come about through the invocation of an interface. The processing that occurs in response to the interface invocation may encounter error conditions, which trigger (through an implementation-specific mechanism) an error message to be generated. In some instances this may be a return value from the interface itself; in other instances a global value may be set and checked after the invocation of an interface. It is likely that a TOE will have a number of low-level error messages that may result from fundamental resource conditions, such as “disk full” or “resource locked”. While these error messages may map to a large number of TSFI, they could be used to detect instances where detail from an interface description has been omitted. For instance, a TSFI that produces a “disk full” message, but has no obvious description of why that TSFI should cause an access to the disk in its description of actions,

5898	might cause the evaluator to examine other evidence (Security Architecture (ADV_ARC),
5899	Objectives
5900	11.4.3.8 The objectives of this sub-activity are to determine whether the formal security policy model of
5901	the TSF clearly and consistently describes the rules and characteristics of the security policies
5902	and whether this description corresponds with the description of security functions in the
5903	functional specification.
5904	11.4.3.8 Input
5905	11.4.3.8 The evaluation evidence for this sub-activity is:
5906	11.4.3.8 the ST;
5907	11.4.3.8 the functional specification;
5908	11.4.3.8 formal security policy model (ADV_SPM.1.1D);
5909	11.4.3.8 formal proof of correspondence between the model and any formal functional specification
5910	(ADV_SPM.1.3D);
5911	11.4.3.8 demonstration of correspondence between the model and the functional specification
5912	(ADV_SPM.1.4D).
5913	11.4.3.8 Application notes
5914	11.4.3.8 This activity applies to cases where the developer has provided a formal security policy model of
5915	the TOE.
5916	11.4.3.8 A formal TOE security policy model is a representation of the rules (synonymously termed
5917	“principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
5918	Their formal counterparts are called security properties and security features, respectively. The
5919	representation includes but is not limited to algebraic specifications, finite state machines and
5920	logic formalisms strong enough to formally infer the properties from the features. The formal TSP
5921	model is accompanied by an informal interpretation explaining how the rules and characteristics
5922	are mapped to the respective properties and features.
5923	11.4.3.8 The creation of a formal security policy model helps to identify and eliminate ambiguous,
5924	inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
5925	built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
5926	of how well the developer has understood the security functionality being implemented and
5927	whether there are inconsistencies between the security requirements and the TOE design. The
5928	confidence in the model is accompanied by a proof that it contains no inconsistencies.
5929	11.4.3.8 A formal security model is a precise formal presentation of the important aspects of security and
5930	their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that
5931	defines the TOE security policy and the set of practises (characteristics) that regulates how the
5932	TSF manages, protects, and otherwise controls the system resources. The model includes the set
5933	of restrictions and properties that specify how information and computing resources are
5934	prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
5935	arguments showing that these restrictions and properties play a key role in the enforcement of
5936	the SFRs. It consists both of the formalisms that express the security functionality, as well as
5937	ancillary text to explain the model and to provide it with context. The security behaviour of the
5938	TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
5939	the TOE and with its operational environment), as well as its internal behaviour.
5940	11.4.3.8 The Security Policy Model of the TOE is informally abstracted from its realisation by considering
5941	the proposed security requirements of the ST. The informal abstraction is taken to be successful if

5942 the TOE's principles turn out to be enforced by its characteristics. The purpose of formal methods
5943 lies within the enhancement of the rigour of enforcement. Informal arguments are always prone
5944 to fallacies; especially if relationships among subjects, objects and operations get more and more
5945 involved. In order to minimise the risk of insecure state arrivals the rules and characteristics of
5946 the security policy model are mapped to respective properties and features within some formal
5947 system, whose rigour and strength can afterwards be used to obtain the security properties by
5948 means of theorems and formal proof.

5949 **11.4.3.8** While the term “formal security policy model” is used in academic circles, the CC's
5950 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
5951 claimed. Therefore, the formal security policy model is merely a formal representation of the set
5952 of SFRs being claimed.

5953 **11.4.3.8** The term security policy has traditionally been associated with only access control
5954 policies, whether label-based (mandatory access control) or user-based (discretionary access
5955 control). However, a security policy is not limited to access control; there are also audit policies,
5956 identification policies, authentication policies, encryption policies, management policies, and any
5957 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
5958 contains an assignment for identifying these policies that are formally modelled.

5959 **11.4.3.8** It is recognized that not all policies can be formally modelled for all TOEs. This is
5960 because either a given policy can not be formally modelled in the otherwise well suited
5961 framework, or because the nature of the TOE renders impossible the modelling of policies that
5962 would otherwise be possible to model.

5963 **11.4.3.8 Action ADV_SPM.1.1E**

5964 **11.4.3.8 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
5965 *text as required, and identify the security policies of the TSF that are modelled.*

5966 **11.4.3.8 Work unit ADV_SPM.1-1**

5967 **11.4.3.8** The evaluator ***shall examine the TOE security policy model to determine that it is***
5968 **written in a formal style.**

5969 **11.4.3.8** The evaluator identifies the formal framework upon which the TOE security policy
5970 model is based and ensures that it is founded on well established mathematical concepts. **They**
5971 **also identify the security properties and features addressed in the application notes and ensure**
5972 **the formalization of at least one security policy.**

5973 **11.4.3.8** For guidance on formal methods refer to ISO/IEC 15408-3

5974 **11.4.3.8 Work unit ADV_SPM.1-2**

5975 **11.4.3.8** The evaluator ***shall examine the TOE security policy model to determine that it***
5976 **contains all necessary informal explanatory text.**

5977 **11.4.3.8** Supporting narrative descriptions are necessary for all parts of the model (for example,
5978 to make clear the meaning of any formal notation and how they are used) including the security
5979 properties and features.

5980 **11.4.3.8 Work unit ADV_SPM.1-3**

5981 **11.4.3.8** The evaluator ***shall examine the TOE security policy model to determine that all***
5982 **security policies of the TSF are identified that are modelled.**

- 5983 **11.4.3.8** The evaluator determines whether the SPM identifies the security policies for which a
 5984 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 5985 of the modelled policies.
- 5986 **11.4.3.8** The evaluator determines whether the list of security policies identified by the SPM is
 5987 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 5988 **11.4.3.8** The evaluator determines whether for each security policy identified by the SPM a
 5989 model is in fact provided.
- 5990 **11.4.3.8 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 5991 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 5992 *not secure.*
- 5993 **11.4.3.8 Work unit ADV_SPM.1-4**
- 5994 **11.4.3.8** The evaluator *shall examine the principles and characteristics of the security policies*
 5995 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 5996 **11.4.3.8** The security policies are expressed in terms of security principles (rules) which are
 5997 modelled by security properties and define the secure state of the TOE. For example, a model
 5998 based on state transitions could describe the security policies in terms of principles of its states,
 5999 identify its initial state, and define what it means to be a secure state.
- 6000 **11.4.3.8** The evaluator determines that the security policies are reflected within their formal
 6001 counterparts of the TSP model.
- 6002 **11.4.3.8** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 6003 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 6004 resources including attributes and conditions of the TOE) which are modelled by security
 6005 features. For example, a model based on state transitions could describe the characteristics as
 6006 possible actions in each secure state in a level of detail sufficient to decide into which state the
 6007 TOE will be transformed by that action.
- 6008 **11.4.3.8** Together the security principles and characteristics describe the entire security posture
 6009 of the TOE.
- 6010 **11.4.3.8** In the context of a formal TOE security policy model the security behaviour is
 6011 considered to be clearly articulated only if an adequate mapping from principles and
 6012 characteristics to their respective formal counterparts properties and features has been given.
 6013 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 6014 detailed enough to allow for correct identification of all security objectives and the relation to the
 6015 security environment.
- 6016 **11.4.3.8** The above condition for clear articulation is necessary but not sufficient. An informal
 6017 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 6018 must be provided in order to make clear their intended meaning.
- 6019 **11.4.3.8 Work unit ADV_SPM.1-5**
- 6020 **11.4.3.8** The evaluator *shall examine the TOE security policy model rationale to determine that*
 6021 *it formally proves that the security features enforce the security properties.*
- 6022 **11.4.3.8** To determine the enforcement, the evaluator considers the security properties and the
 6023 security features and verifies that the arguments used in the proof are valid. The proof of
 6024 correspondence between the security properties and the security features shall be formal.

- 6025 **11.4.3.8** The validity of the security properties shall mean that the TOE is in a secure state. By
 6026 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 6027 state.
- 6028 **11.4.3.8 Work unit ADV_SPM.1-6**
- 6029 **11.4.3.8** The evaluator *shall examine the TOE security policy model rationale to determine that*
 6030 it proves the internal consistency of the TOE security policy model.
- 6031 **11.4.3.8** The proof shall show the absence of contradictions within the TOE security policy
 6032 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 6033 used in the proof are valid.
- 6034 **11.4.3.8** Since the TOE security policy model is formal, the proof of its internal consistency shall
 6035 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 6036 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 6037 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 6038 security policy model that prove the internal consistency by means of a combination with generic
 6039 arguments of the formal framework.
- 6040 **11.4.3.8 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 6041 *specification shall be at the correct level of formality.*
- 6042 **11.4.3.8 Work unit ADV_SPM.1-7**
- 6043 **11.4.3.8** The evaluator *shall examine the correspondence between the model and the functional*
 6044 *specification to determine that a semiformal demonstration of correspondence between the*
 6045 *model and any semiformal functional specification is provided.*
- 6046 **11.4.3.8** This work unit is only applicable to a semiformal presentation of the functional
 6047 specification, which is required by ADV_FSP.5.2C.
- 6048 **11.4.3.8** A semiformal correspondence is one that results from a structured approach with a
 6049 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 6050 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 6051 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 6052 **11.4.3.8** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 6053 **methods**'.
- 6054 **11.4.3.8 Work unit ADV_SPM.1-8**
- 6055 **11.4.3.8** The evaluator *shall examine the correspondence between the model and the functional*
 6056 *specification to determine that a formal proof of correspondence between the model and any*
 6057 *formal functional specification is provided.*
- 6058 **11.4.3.8** This work unit is only applicable to a formal presentation of the functional specification,
 6059 which is required by ADV_FSP.6.2D.
- 6060 **11.4.3.8** There should be a formal proof of correspondence between the model and any formal
 6061 functional specification.
- 6062 **11.4.3.8** The formal proof of correspondence removes all subjective interpretations of its terms
 6063 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 6064 formal notation and uses rules that support logical reasoning. The security features within the
 6065 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 6066 language and shown to be satisfied by the formal specification.

- 6067 **11.4.3.8** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 6068 **11.4.3.8 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 6069 *specification is consistent and complete with respect to the model.*
- 6070 **11.4.3.8 Work unit ADV_SPM.1-9**
- 6071 **11.4.3.8** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 6072 ***TSF interfaces (as articulated in the functional specification) is complete with respect to the***
 6073 ***behaviour modelled by the security features.***
- 6074 **11.4.3.8** The term “correspondence” here means both the formal proof of correspondence
 6075 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 6076 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 6077 **11.4.3.8** In determining completeness of the correspondence, the evaluator considers the
 6078 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 6079 features of the TSP model. The demonstration should show that all characteristics belonging to
 6080 policies that are required to be modelled have an associated feature description in the TOE
 6081 security policy model, and that each feature of the TSP model does occur in the mapping.
- 6082 **11.4.3.8** Abstention from formally modelling TSFI behaviour always calls for justification on the
 6083 developer’s side (also confer the application notes above).
- 6084 **11.4.3.8 Work unit ADV_SPM.1-10**
- 6085 **11.4.3.8** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 6086 ***TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
 6087 ***behaviour modelled by the security features.***
- 6088 **11.4.3.8** The term “correspondence” here means both the formal proof of correspondence
 6089 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 6090 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 6091 **11.4.3.8** The meaning of consistency reflects the conventional understanding in contrast to the
 6092 internal consistency concept of work unit ADV_SPM.1-6.
- 6093 **11.4.3.8** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 6094 security features established in the preceding work unit and verifies that the correspondence
 6095 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 6096 behaviour.
- 6097 **11.4.3.8** For example, if TSFI behaviour dealt with access management on the granularity of
 6098 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 6099 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 6100 management for groups of users, then a TSP model describing the security behaviour of the TOE
 6101 in terms of individual users would also not be consistent.
- 6102 **11.4.3.8** As another example, if remote untrusted users had to pass more stringent
 6103 authentication procedures than administrators whose only point of access were within a
 6104 physically-protected area, then this difference in authentication procedures had to be reflected in
 6105 the security features.
- 6106 **11.4.3.8** TOE design (ADV_TDS)) related that TSFI to determine if the description is accurate.
- 6107 In order to determine that the description of the error messages of a TSFI is accurate and complete,
 6108 the evaluator measures the interface description against the other evidence provided for the
 6109 evaluation (e.g., TOE design, security architecture description, operational user guidance), as well

6110 as for other evidence supplied for that TSFI (description of SFR-enforcing actions, summary of SFR-
6111 supporting and SFR-non-interfering actions and results).

6112 ISO/IEC 15408-3 ADV_FSP.3.6C: *The functional specification shall summarise the SFR-supporting*
6113 *and SFR-non-interfering actions associated with each TSFI.*

6114 **11.4.3.11.11 Work unit ADV_FSP.3-8**

6115 The evaluator **shall examine** the presentation of the TSFI to determine that it summarises the SFR-
6116 supporting and SFR-non-interfering actions associated with each TSFI.

6117 The purpose of this work unit is to supplement the details about the SFR-enforcing actions
6118 (provided in work unit ADV_FSP.3-6) with a summary of the remaining actions (i.e., those that are
6119 not SFR-enforcing). This covers *all* SFR-supporting and SFR-non-interfering actions, whether
6120 invokable through SFR-enforcing TSFI or through SFR-supporting or SFR-non-interfering TSFI.
6121 Such a summary about all SFR-supporting and SFR-non-interfering actions helps to provide a more
6122 complete picture of the functions provided by the TSF, and is to be used by the evaluator in
6123 determining whether an action or TSFI may have been mis-categorised.

6124 The information to be provided is more abstract than that required for SFR-enforcing actions.
6125 While it should still be detailed enough so that the reader can understand what the action does, the
6126 description does not have to be detailed enough to support writing tests against it, for instance. For
6127 the evaluator, the key is that the information must be sufficient to make a positive determination
6128 that the action is SFR-supporting or SFR-non-interfering. If that level of information is missing, the
6129 summary is insufficient and more information must be obtained.

6130 ISO/IEC 15408-3 ADV_FSP.3.7C: *The tracing shall demonstrate that the SFRs trace to TSFIs in the*
6131 *functional specification.*

6132 **11.4.3.11.12 Work unit ADV_FSP.3-9**

6133 The evaluator **shall check** that the tracing links the SFRs to the corresponding TSFIs.

6134 The tracing is provided by the developer to serve as a guide to which SFRs are related to which
6135 TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the
6136 following work units, in which the evaluator verifies its completeness and accuracy.

6137 **11.4.3.12 Action ADV_FSP.3.2E**

6138 **11.4.3.12.1 Work unit ADV_FSP.3-10**

6139 The evaluator **shall examine** the functional specification to determine that it is a complete
6140 instantiation of the SFRs.

6141 To ensure that all SFRs are covered by the functional specification, as well as the test coverage
6142 analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.3-9 a map between
6143 the TOE security functional requirements and the TSFI. Note that this map may have to be at a level
6144 of detail below the component or even element level of the requirements, because of operations
6145 (assignments, refinements, selections) performed on the functional requirement by the ST author.

6146 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained,
6147 for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three
6148 different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and
6149 claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to
6150 TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper
6151 interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of
6152 parameters for a given interface.

6153 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 6154 boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the
 6155 TSFI. The analysis for those requirements will be performed in the analysis for the TOE design
 6156 (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions,
 6157 and error messages associated with TSFIs must be fully specified, the evaluator should be able to
 6158 determine if all aspects of an SFR appear to be implemented at the interface level.

6159 **11.4.3.12.2 Work unit ADV_FSP.3-11**

6160 The evaluator ***shall examine*** the functional specification to determine that it is an accurate
 6161 instantiation of the SFRs.

6162 For each functional requirement in the ST that results in effects visible at the TSF boundary, the
 6163 information in the associated TSFI for that requirement specifies the required functionality
 6164 described by the requirement. For example, if the ST contains a requirement for access control lists,
 6165 and the only TSFI that map to that requirement specify functionality for Unix-style protection bits,
 6166 then the functional specification is not accurate with respect to the requirements.

6167 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 6168 boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements
 6169 to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE
 6170 design (ADV_TDS) when included in the ST.

6171 **11.4.4 Evaluation of sub-activity (ADV_FSP.4)**

6172 **11.4.4.1 Objectives**

6173 The objective of this sub-activity is to determine whether the developer has completely described
 6174 all of the TSFI in a manner such that the evaluator is able to determine whether the TSFI are
 6175 completely and accurately described, and appears to implement the security functional
 6176 requirements of the ST.

6177 **11.4.4.2 Input**

6178 The evaluation evidence for this sub-activity that is required by the work-units is:

- 6179 a) the ST;
- 6180 b) the functional specification;
- 6181 c) the TOE design.

6182 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

- 6183 a) the security architecture description;
- 6184 b) the implementation representation;
- 6185 c) the TSF internals description;
- 6186 d) the operational user guidance;

6187 **11.4.4.3 Application notes**

6188 The functional specification describes the interfaces to the TSF (the TSFI) in a structured manner.
 6189 Because of the dependency on Evaluation of sub-activity (ADV_TDS.1), the evaluator is expected to
 6190 have identified the TSF prior to beginning work on this sub-activity. Without firm knowledge of
 6191 what comprises the TSF, it is not possible to assess the completeness of the TSFI.

6192 In performing the various work units included in this family, the evaluator is asked to make
 6193 assessments of accuracy and completeness of several factors (the TSFI itself, as well as the
 6194 individual components (parameters, actions, error messages, etc.) of the TSFI). In doing this
 6195 analysis, the evaluator is expected to use the documentation provided for the evaluation. This
 6196 includes the ST, the TOE design, and may include other documentation such as the operational user
 6197 guidance, security architecture description, and implementation representation. The
 6198 documentation should be examined in an iterative fashion. The evaluator may read, for example, in
 6199 the TOE design how a certain function is implemented, but see no way to invoke that function from
 6200 the interface. This might cause the evaluator to question the completeness of a particular TSFI
 6201 description, or whether an interface has been left out of the functional specification altogether.
 6202 Describing analysis activities of this sort in the ETR is a key method in providing rationale that the
 6203 work units have been performed appropriately.

6204 It should be recognised that there exist functional requirements whose functionality is manifested
 6205 wholly or in part architecturally, rather than through a specific mechanism. An example of this is
 6206 the implementation of mechanisms implementing the **Residual information protection (FDP_RIP)**
 6207 requirements. Such mechanisms typically are implemented to ensure a behaviour isn't present,
 6208 which is difficult to test and typically is verified through analysis. In the cases where such
 6209 functional requirements are included in the ST, it is expected that the evaluator recognise that
 6210 there may be SFRs of this type that have no interfaces, and that this should not be considered a
 6211 deficiency in the functional specification.

6212 **11.4.4.4 Action ADV_FSP.4.1E**

6213 ISO/IEC 15408-3 ADV_FSP.4.1C: *The functional specification shall completely represent the TSF.*

6214 **11.4.4.4.1 Work unit ADV_FSP.4-1**

6215 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully
 6216 represented.

6217 **11.4.4.5 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity.** 6218 **The TSF must be identified (done as part of the Objectives)**

6219 **11.4.4.5** The objectives of this sub-activity are to determine whether the formal security policy model of
 6220 the TSF clearly and consistently describes the rules and characteristics of the security policies
 6221 and whether this description corresponds with the description of security functions in the
 6222 functional specification.

6223 **11.4.4.5 Input**

6224 **11.4.4.5** The evaluation evidence for this sub-activity is:

6225 **11.4.4.5** the ST;

6226 **11.4.4.5** the functional specification;

6227 **11.4.4.5** formal security policy model (ADV_SPM.1.1D);

6228 **11.4.4.5** formal proof of correspondence between the model and any formal functional specification
 6229 (ADV_SPM.1.3D);

6230 **11.4.4.5** demonstration of correspondence between the model and the functional specification
 6231 (ADV_SPM.1.4D).

6232 **11.4.4.5 Application notes**

6233 **11.4.4.5** This activity applies to cases where the developer has provided a formal security policy
6234 model of the TOE.

6235 **11.4.4.5** A formal TOE security policy model is a representation of the rules (synonymously
6236 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
6237 terms. Their formal counterparts are called security properties and security features,
6238 respectively. The representation includes but is not limited to algebraic specifications, finite state
6239 machines and logic formalisms strong enough to formally infer the properties from the features.
6240 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
6241 characteristics are mapped to the respective properties and features.

6242 **11.4.4.5** The creation of a formal security policy model helps to identify and eliminate
6243 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
6244 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
6245 judgement of how well the developer has understood the security functionality being
6246 implemented and whether there are inconsistencies between the security requirements and the
6247 TOE design. The confidence in the model is accompanied by a proof that it contains no
6248 inconsistencies.

6249 **11.4.4.5** A formal security model is a precise formal presentation of the important aspects of
6250 security and their relationship to the behaviour of the TOE; it identifies the set of rules
6251 (principles) that defines the TOE security policy and the set of practises (characteristics) that
6252 regulates how the TSF manages, protects, and otherwise controls the system resources. The
6253 model includes the set of restrictions and properties that specify how information and computing
6254 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
6255 engineering arguments showing that these restrictions and properties play a key role in the
6256 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
6257 as well as ancillary text to explain the model and to provide it with context. The security
6258 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
6259 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

6260 **11.4.4.5** The Security Policy Model of the TOE is informally abstracted from its realisation by
6261 considering the proposed security requirements of the ST. The informal abstraction is taken to be
6262 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
6263 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
6264 are always prone to fallacies; especially if relationships among subjects, objects and operations
6265 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
6266 characteristics of the security policy model are mapped to respective properties and features
6267 within some formal system, whose rigour and strength can afterwards be used to obtain the
6268 security properties by means of theorems and formal proof.

6269 **11.4.4.5** While the term “formal security policy model” is used in academic circles, the CC's
6270 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
6271 claimed. Therefore, the formal security policy model is merely a formal representation of the set
6272 of SFRs being claimed.

6273 **11.4.4.5** The term security policy has traditionally been associated with only access control
6274 policies, whether label-based (mandatory access control) or user-based (discretionary access
6275 control). However, a security policy is not limited to access control; there are also audit policies,
6276 identification policies, authentication policies, encryption policies, management policies, and any
6277 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
6278 contains an assignment for identifying these policies that are formally modelled.

6279 **11.4.4.5** It is recognized that not all policies can be formally modelled for all TOEs. This is
6280 because either a given policy can not be formally modelled in the otherwise well suited

- 6281 framework, or because the nature of the TOE renders impossible the modelling of policies that
6282 would otherwise be possible to model.
- 6283 **11.4.4.5 Action ADV_SPM.1.1E**
- 6284 **11.4.4.5 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
6285 *text as required, and identify the security policies of the TSF that are modelled.*
- 6286 **11.4.4.5 Work unit ADV_SPM.1-1**
- 6287 **11.4.4.5** The evaluator ***shall examine the TOE security policy model to determine that it is***
6288 ***written in a formal style.***
- 6289 **11.4.4.5** The evaluator identifies the formal framework upon which the TOE security policy
6290 model is based and ensures that it is founded on well established mathematical concepts. **They**
6291 **also identify the security properties and features addressed in the application notes and ensure**
6292 **the formalization of at least one security policy.**
- 6293 **11.4.4.5** For guidance on formal methods refer to ISO/IEC 15408-3
- 6294 **11.4.4.5 Work unit ADV_SPM.1-2**
- 6295 **11.4.4.5** The evaluator ***shall examine the TOE security policy model to determine that it***
6296 ***contains all necessary informal explanatory text.***
- 6297 **11.4.4.5** Supporting narrative descriptions are necessary for all parts of the model (for example,
6298 to make clear the meaning of any formal notation and how they are used) including the security
6299 properties and features.
- 6300 **11.4.4.5 Work unit ADV_SPM.1-3**
- 6301 **11.4.4.5** The evaluator ***shall examine the TOE security policy model to determine that all***
6302 ***security policies of the TSF are identified that are modelled.***
- 6303 **11.4.4.5** The evaluator determines whether the SPM identifies the security policies for which a
6304 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
6305 of the modelled policies.
- 6306 **11.4.4.5** The evaluator determines whether the list of security policies identified by the SPM is
6307 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 6308 **11.4.4.5** The evaluator determines whether for each security policy identified by the SPM a
6309 model is in fact provided.
- 6310 **11.4.4.5 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
6311 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
6312 *not secure.*
- 6313 **11.4.4.5 Work unit ADV_SPM.1-4**
- 6314 **11.4.4.5** The evaluator ***shall examine the principles and characteristics of the security policies***
6315 ***to determine that the modelled security behaviour of the TOE is clearly articulated.***
- 6316 **11.4.4.5** The security policies are expressed in terms of security principles (rules) which are
6317 modelled by security properties and define the secure state of the TOE. For example, a model
6318 based on state transitions could describe the security policies in terms of principles of its states,
6319 identify its initial state, and define what it means to be a secure state.

- 6320 **11.4.4.5** The evaluator determines that the security policies are reflected within their formal
6321 counterparts of the TSP model.
- 6322 **11.4.4.5** The TOE security behaviour is expressed in terms of security characteristics (i.e.
6323 portions of TOE security functionality managing, protecting, and otherwise controlling the system
6324 resources including attributes and conditions of the TOE) which are modelled by security
6325 features. For example, a model based on state transitions could describe the characteristics as
6326 possible actions in each secure state in a level of detail sufficient to decide into which state the
6327 TOE will be transformed by that action.
- 6328 **11.4.4.5** Together the security principles and characteristics describe the entire security posture
6329 of the TOE.
- 6330 **11.4.4.5** In the context of a formal TOE security policy model the security behaviour is
6331 considered to be clearly articulated only if an adequate mapping from principles and
6332 characteristics to their respective formal counterparts properties and features has been given.
6333 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
6334 detailed enough to allow for correct identification of all security objectives and the relation to the
6335 security environment.
- 6336 **11.4.4.5** The above condition for clear articulation is necessary but not sufficient. An informal
6337 interpretation of all formal concepts (including attributes, predicates and variables, if available)
6338 must be provided in order to make clear their intended meaning.
- 6339 **11.4.4.5 Work unit ADV_SPM.1-5**
- 6340 **11.4.4.5** The evaluator ***shall examine the TOE security policy model rationale to determine that***
6341 *it formally proves that the security features enforce the security properties.*
- 6342 **11.4.4.5** To determine the enforcement, the evaluator considers the security properties and the
6343 security features and verifies that the arguments used in the proof are valid. The proof of
6344 correspondence between the security properties and the security features shall be formal.
- 6345 **11.4.4.5** The validity of the security properties shall mean that the TOE is in a secure state. By
6346 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
6347 state.
- 6348 **11.4.4.5 Work unit ADV_SPM.1-6**
- 6349 **11.4.4.5** The evaluator ***shall examine the TOE security policy model rationale to determine that***
6350 *it proves the internal consistency of the TOE security policy model.*
- 6351 **11.4.4.5** The proof shall show the absence of contradictions within the TOE security policy
6352 model. In determining the absence of contradictions, the evaluator verifies that the arguments
6353 used in the proof are valid.
- 6354 **11.4.4.5** Since the TOE security policy model is formal, the proof of its internal consistency shall
6355 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
6356 security policy model usually is not possible due to the fundamental nature of formal frameworks.
6357 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
6358 security policy model that prove the internal consistency by means of a combination with generic
6359 arguments of the formal framework.
- 6360 **11.4.4.5 ADV_SPM.1.3C** ***The correspondence between the model and the functional***
6361 ***specification shall be at the correct level of formality.***

- 6362 **11.4.4.5 Work unit ADV_SPM.1-7**
- 6363 **11.4.4.5** The evaluator *shall examine the correspondence between the model and the*
6364 functional specification to determine that a semiformal demonstration of correspondence
6365 between the model and any semiformal functional specification is provided.
- 6366 **11.4.4.5** This work unit is only applicable to a semiformal presentation of the functional
6367 specification, which is required by ADV_FSP.5.2C.
- 6368 **11.4.4.5** A semiformal correspondence is one that results from a structured approach with a
6369 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
6370 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
6371 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 6372 **11.4.4.5** For guidance on semiformal methods refer to Annex 3.1.1 ‘**Semiformal and formal**
6373 **methods**’.
- 6374 **11.4.4.5 Work unit ADV_SPM.1-8**
- 6375 **11.4.4.5** The evaluator *shall examine the correspondence between the model and the functional*
6376 specification to determine that a formal proof of correspondence between the model and any
6377 formal functional specification is provided.
- 6378 **11.4.4.5** This work unit is only applicable to a formal presentation of the functional specification,
6379 which is required by ADV_FSP.6.2D.
- 6380 **11.4.4.5** There should be a formal proof of correspondence between the model and any formal
6381 functional specification.
- 6382 **11.4.4.5** The formal proof of correspondence removes all subjective interpretations of its terms
6383 by enlisting well-established mathematical concepts to define the syntax and semantics of the
6384 formal notation and uses rules that support logical reasoning. The security features within the
6385 TOE (which are identified in the formal TSP model) are expressed in a formal specification
6386 language and shown to be satisfied by the formal specification.
- 6387 **11.4.4.5** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 6388 **11.4.4.5 ADV_SPM.1.4C** *The correspondence shall show that the functional*
6389 *specification is consistent and complete with respect to the model.*
- 6390 **11.4.4.5 Work unit ADV_SPM.1-9**
- 6391 **11.4.4.5** The evaluator *shall examine the correspondence to determine that the behaviour at the*
6392 TSF interfaces (as articulated in the functional specification) is complete with respect to the
6393 behaviour modelled by the security features.
- 6394 **11.4.4.5** The term “correspondence” here means both the formal proof of correspondence
6395 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
6396 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 6397 **11.4.4.5** In determining completeness of the correspondence, the evaluator considers the
6398 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
6399 features of the TSP model. The demonstration should show that all characteristics belonging to
6400 policies that are required to be modelled have an associated feature description in the TOE
6401 security policy model, and that each feature of the TSP model does occur in the mapping.
- 6402 **11.4.4.5** Abstention from formally modelling TSFI behaviour always calls for justification on the
6403 developer’s side (also confer the application notes above).

6404 **11.4.4.5 Work unit ADV_SPM.1-10**

6405 **11.4.4.5** The evaluator *shall examine* the correspondence to determine that the behaviour at the
6406 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
6407 behaviour modelled by the security features.

6408 **11.4.4.5** The term “correspondence” here means both the formal proof of correspondence
6409 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
6410 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

6411 **11.4.4.5** The meaning of consistency reflects the conventional understanding in contrast to the
6412 internal consistency concept of work unit ADV_SPM.1-6.

6413 **11.4.4.5** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
6414 security features established in the preceding work unit and verifies that the correspondence
6415 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
6416 behaviour.

6417 **11.4.4.5** For example, if TSFI behaviour dealt with access management on the granularity of
6418 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
6419 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
6420 management for groups of users, then a TSP model describing the security behaviour of the TOE
6421 in terms of individual users would also not be consistent.

6422 **11.4.4.5** As another example, if remote untrusted users had to pass more stringent
6423 authentication procedures than administrators whose only point of access were within a
6424 physically-protected area, then this difference in authentication procedures had to be reflected in
6425 the security features.

6426 **11.4.4.5** TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be
6427 done at a high level to ensure that no large groups of interfaces have been missed (network
6428 protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the
6429 functional specification proceeds.

6430 In making an assessment for this work unit, the evaluator determines that all portions of the TSF
6431 are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF
6432 should have a corresponding interface description, or if there are no corresponding interfaces for a
6433 portion of the TSF, the evaluator determines that that is acceptable.

6434 ISO/IEC 15408-3 ADV_FSP.4.2C: *The functional specification shall describe the purpose and method*
6435 *of use for all TSFI.*

6436 **11.4.4.8.11 Work unit ADV_FSP.4-2**

6437 The evaluator *shall examine* the functional specification to determine that it states the purpose of
6438 each TSFI.

6439 The purpose of a TSFI is a general statement summarising the functionality provided by the
6440 interface. It is not intended to be a complete statement of the actions and results related to the
6441 interface, but rather a statement to help the reader understand in general what the interface is
6442 intended to be used for. The evaluator should not only determine that the purpose exists, but also
6443 that it accurately reflects the TSFI by taking into account other information about the interface,
6444 such as the description of actions and error messages.

6445 **11.4.4.8.12 Work unit ADV_FSP.4-3**

6446 The evaluator *shall examine* the functional specification to determine that the method of use for
6447 each TSFI is given.

6448 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the
 6449 actions and obtain the results associated with the TSFI. The evaluator should be able to determine,
 6450 from reading this material in the functional specification, how to use each interface. This does not
 6451 necessarily mean that there needs to be a separate method of use for each TSFI, as it may be
 6452 possible to describe in general how kernel calls are invoked, for instance, and then identify each
 6453 interface using that general style. Different types of interfaces will require different method of use
 6454 specifications. APIs, network protocol interfaces, system configuration parameters, and hardware
 6455 bus interfaces all have very different methods of use, and this should be taken into account by the
 6456 developer when developing the functional specification, as well as by the evaluator evaluating the
 6457 functional specification.

6458 For administrative interfaces whose functionality is documented as being inaccessible to untrusted
 6459 users, the evaluator ensures that the method of making the functions inaccessible is described in
 6460 the functional specification. It should be noted that this inaccessibility needs to be tested by the
 6461 developer in their test suite.

6462 The evaluator should not only determine that the set of method of use descriptions exist, but also
 6463 that they accurately cover each TSFI.

6464 **11.4.4.8.13 Work unit ADV_FSP.4-4**

6465 The evaluator ***shall examine*** the functional specification to determine the completeness of the
 6466 TSFI

6467 The evaluator shall use the design documentation to identify the possible types of interfaces. The
 6468 evaluator shall search the design documentation and the guidance documentation for potential
 6469 TSFI not contained in the developer's documentation, thus indicating that the set of TSFI defined
 6470 by the developer is incomplete. The evaluator ***shall examine*** the arguments presented by the
 6471 developer that the TSFI is complete and check down to the lowest level of design or with the
 6472 implementation representation that no additional TSFI exist.

6473 ISO/IEC 15408-3 ADV_FSP.4.3C: *The functional specification shall identify and describe all*
 6474 *parameters associated with each TSFI.*

6475 **11.4.4.8.14 Work unit ADV_FSP.4-5**

6476 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies
 6477 all parameters associated with every TSFI.

6478 The evaluator examines the functional specification to ensure that all of the parameters are
 6479 described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the
 6480 behaviour of that interface. For examples, parameters are the arguments supplied to an API; the
 6481 various fields in packet for a given network protocol; the individual key values in the Windows
 6482 Registry; the signals across a set of pins on a chip; etc.

6483 In order to determine that all of the parameters are present in the TSFI, the evaluator should
 6484 examine the rest of the interface description (actions, error messages, etc.) to determine if the
 6485 effects of the parameter are accounted for in the description. The evaluator should also check other
 6486 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 6487 operational user guidance, implementation representation) to see if behaviour or additional
 6488 parameters are described there but not in the functional specification.

6489 **11.4.4.8.15 Work unit ADV_FSP.4-6**

6490 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 6491 accurately describes all parameters associated with every TSFI.

6492 Once all of the parameters have been identified, the evaluator needs to ensure that they are
 6493 accurately described, and that the description of the parameters is complete. A parameter
 6494 description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)*
 6495 could be described as having “parameter *i* which is an integer”; this is not an acceptable parameter
 6496 description. A description such as “parameter *i* is an integer that indicates the number of users
 6497 currently logged in to the system” is much more acceptable.

6498 In order to determine that the description of the parameters is complete, the evaluator should
 6499 examine the rest of the interface description (purpose, method of use, actions, error messages, etc.)
 6500 to determine if the descriptions of the parameter(s) are accounted for in the description. The
 6501 evaluator should also check other evidence provided (e.g., TOE design, architectural design,
 6502 operational user guidance, implementation representation) to see if behaviour or additional
 6503 parameters are described there but not in the functional specification.

6504 ISO/IEC 15408-3 ADV_FSP.4.4C: *The functional specification shall describe all actions associated*
 6505 *with each TSFI.*

6506 **11.4.4.8.16 Work unit ADV_FSP.4-7**

6507 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 6508 accurately describes all actions associated with every TSFI.

6509 The evaluator checks to ensure that all of the actions are described. actions available through an
 6510 interface describe what the interface does (as opposed to the TOE design, which describes how the
 6511 actions are provided by the TSF).

6512 Actions of an interface describe functionality that can be invoked through the interface, and can be
 6513 categorised as *regular* actions, and *SFR-related* actions. Regular actions are descriptions of what the
 6514 interface does. The amount of information provided for this description is dependant on the
 6515 complexity of the interface. The SFR-related actions are those that are visible at any external
 6516 interface (for instance, audit activity caused by the invocation of an interface (assuming audit
 6517 requirements are included in the ST) should be described, even though the result of that action is
 6518 generally not visible through the invoked interface). Depending on the parameters of an interface,
 6519 there may be many different actions able to be invoked through the interface (for instance, an API
 6520 might have the first parameter be a “subcommand”, and the following parameters be specific to
 6521 that subcommand. The IOCTL API in some Unix systems is an example of such an interface).

6522 In order to determine that the description of the actions of a TSFI is complete, the evaluator should
 6523 review the rest of the interface description (parameter descriptions, error messages, etc.) to
 6524 determine if the actions described are accounted for. The evaluator should also analyse other
 6525 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 6526 operational user guidance, implementation representation) to see if there is evidence of actions
 6527 that are described there but not in the functional specification.

6528 ISO/IEC 15408-3 ADV_FSP.4.5C: *The functional specification shall describe all direct error messages*
 6529 *that may result from an invocation of each TSFI.*

6530 **11.4.4.8.17 Work unit ADV_FSP.4-8**

6531 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 6532 accurately describes all errors messages resulting from an invocation of each TSFI.

6533 Errors can take many forms, depending on the interface being described. For an API, the interface
 6534 itself may return an error code; set a global error condition, or set a certain parameter with an
 6535 error code. For a configuration file, an incorrectly configured parameter may cause an error
 6536 message to be written to a log file. For a hardware PCI card, an error condition may raise a signal
 6537 on the bus, or trigger an exception condition to the CPU.

11.4.4.9 Errors (and the associated error messages) come about through the invocation of an interface. The processing that occurs in response to the interface invocation may encounter error conditions, which trigger (through an implementation-specific mechanism) an error message to be generated. In some instances this may be a return value from the interface itself; in other instances a global value may be set and checked after the invocation of an interface. It is likely that a TOE will have a number of low-level error messages that may result from fundamental resource conditions, such as “disk full” or “resource locked”. While these error messages may map to a large number of TSFI, they could be used to detect instances where detail from an interface description has been omitted. For instance, a TSFI that produces a “disk full” message, but has no obvious description of why that TSFI should cause an access to the disk in its description of actions, might cause the evaluator to examine other evidence (Security Architecture (ADV_ARC), Objectives

11.4.4.9 The objectives of this sub-activity are to determine whether the formal security policy model of the TSF clearly and consistently describes the rules and characteristics of the security policies and whether this description corresponds with the description of security functions in the functional specification.

11.4.4.9 Input

11.4.4.9 The evaluation evidence for this sub-activity is:

11.4.4.9 the ST;

11.4.4.9 the functional specification;

11.4.4.9 formal security policy model (ADV_SPM.1.1D);

11.4.4.9 formal proof of correspondence between the model and any formal functional specification (ADV_SPM.1.3D);

11.4.4.9 demonstration of correspondence between the model and the functional specification (ADV_SPM.1.4D).

11.4.4.9 Application notes

11.4.4.9 This activity applies to cases where the developer has provided a formal security policy model of the TOE.

11.4.4.9 A formal TOE security policy model is a representation of the rules (synonymously termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms. Their formal counterparts are called security properties and security features, respectively. The representation includes but is not limited to algebraic specifications, finite state machines and logic formalisms strong enough to formally infer the properties from the features. The formal TSP model is accompanied by an informal interpretation explaining how the rules and characteristics are mapped to the respective properties and features.

11.4.4.9 The creation of a formal security policy model helps to identify and eliminate ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been built, the formal model serves the evaluation effort by contributing to the evaluator's judgement of how well the developer has understood the security functionality being implemented and whether there are inconsistencies between the security requirements and the TOE design. The confidence in the model is accompanied by a proof that it contains no inconsistencies.

11.4.4.9 A formal security model is a precise formal presentation of the important aspects of security and their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that defines the TOE security policy and the set of practises (characteristics) that regulates how the

6583 TSF manages, protects, and otherwise controls the system resources. The model includes the set
 6584 of restrictions and properties that specify how information and computing resources are
 6585 prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
 6586 arguments showing that these restrictions and properties play a key role in the enforcement of
 6587 the SFRs. It consists both of the formalisms that express the security functionality, as well as
 6588 ancillary text to explain the model and to provide it with context. The security behaviour of the
 6589 TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
 6590 the TOE and with its operational environment), as well as its internal behaviour.

6591 **11.4.4.9** The Security Policy Model of the TOE is informally abstracted from its realisation by
 6592 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 6593 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 6594 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 6595 are always prone to fallacies; especially if relationships among subjects, objects and operations
 6596 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 6597 characteristics of the security policy model are mapped to respective properties and features
 6598 within some formal system, whose rigour and strength can afterwards be used to obtain the
 6599 security properties by means of theorems and formal proof.

6600 **11.4.4.9** While the term "formal security policy model" is used in academic circles, the CC's
 6601 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
 6602 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 6603 of SFRs being claimed.

6604 **11.4.4.9** The term security policy has traditionally been associated with only access control
 6605 policies, whether label-based (mandatory access control) or user-based (discretionary access
 6606 control). However, a security policy is not limited to access control; there are also audit policies,
 6607 identification policies, authentication policies, encryption policies, management policies, and any
 6608 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 6609 contains an assignment for identifying these policies that are formally modelled.

6610 **11.4.4.9** It is recognized that not all policies can be formally modelled for all TOEs. This is
 6611 because either a given policy can not be formally modelled in the otherwise well suited
 6612 framework, or because the nature of the TOE renders impossible the modelling of policies that
 6613 would otherwise be possible to model.

6614 **11.4.4.9 Action ADV_SPM.1.1E**

6615 **11.4.4.9 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 6616 *text as required, and identify the security policies of the TSF that are modelled.*

6617 **11.4.4.9 Work unit ADV_SPM.1-1**

6618 **11.4.4.9** The evaluator *shall examine* the TOE security policy model to determine that it is
 6619 written in a formal style.

6620 **11.4.4.9** The evaluator identifies the formal framework upon which the TOE security policy
 6621 model is based and ensures that it is founded on well established mathematical concepts. **They**
 6622 also identify **the security properties and features addressed in the application notes and ensure**
 6623 the formalization of at least one security policy.

6624 **11.4.4.9** For guidance on formal methods refer to ISO/IEC 15408-3

6625 **11.4.4.9 Work unit ADV_SPM.1-2**

6626 **11.4.4.9** The evaluator *shall examine* the TOE security policy model to determine that it
 6627 contains all necessary informal explanatory text.

- 6628 **11.4.4.9** Supporting narrative descriptions are necessary for all parts of the model (for example,
6629 to make clear the meaning of any formal notation and how they are used) including the security
6630 properties and features.
- 6631 **11.4.4.9 Work unit ADV_SPM.1-3**
- 6632 **11.4.4.9** The evaluator *shall examine the TOE security policy model to determine that all*
6633 security policies of the TSF are identified that are modelled.
- 6634 **11.4.4.9** The evaluator determines whether the SPM identifies the security policies for which a
6635 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
6636 of the modelled policies.
- 6637 **11.4.4.9** The evaluator determines whether the list of security policies identified by the SPM is
6638 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 6639 **11.4.4.9** The evaluator determines whether for each security policy identified by the SPM a
6640 model is in fact provided.
- 6641 **11.4.4.9 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
6642 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
6643 *not secure.*
- 6644 **11.4.4.9 Work unit ADV_SPM.1-4**
- 6645 **11.4.4.9** The evaluator *shall examine the principles and characteristics of the security policies*
6646 to determine that the modelled security behaviour of the TOE is clearly articulated.
- 6647 **11.4.4.9** The security policies are expressed in terms of security principles (rules) which are
6648 modelled by security properties and define the secure state of the TOE. For example, a model
6649 based on state transitions could describe the security policies in terms of principles of its states,
6650 identify its initial state, and define what it means to be a secure state.
- 6651 **11.4.4.9** The evaluator determines that the security policies are reflected within their formal
6652 counterparts of the TSP model.
- 6653 **11.4.4.9** The TOE security behaviour is expressed in terms of security characteristics (i.e.
6654 portions of TOE security functionality managing, protecting, and otherwise controlling the system
6655 resources including attributes and conditions of the TOE) which are modelled by security
6656 features. For example, a model based on state transitions could describe the characteristics as
6657 possible actions in each secure state in a level of detail sufficient to decide into which state the
6658 TOE will be transformed by that action.
- 6659 **11.4.4.9** Together the security principles and characteristics describe the entire security posture
6660 of the TOE.
- 6661 **11.4.4.9** In the context of a formal TOE security policy model the security behaviour is
6662 considered to be clearly articulated only if an adequate mapping from principles and
6663 characteristics to their respective formal counterparts properties and features has been given.
6664 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
6665 detailed enough to allow for correct identification of all security objectives and the relation to the
6666 security environment.
- 6667 **11.4.4.9** The above condition for clear articulation is necessary but not sufficient. An informal
6668 interpretation of all formal concepts (including attributes, predicates and variables, if available)
6669 must be provided in order to make clear their intended meaning.

- 6670 **11.4.4.9 Work unit ADV_SPM.1-5**
- 6671 **11.4.4.9** The evaluator *shall examine the TOE security policy model rationale to determine*
6672 *that it formally proves that the security features enforce the security properties.*
- 6673 **11.4.4.9** To determine the enforcement, the evaluator considers the security properties and the
6674 security features and verifies that the arguments used in the proof are valid. The proof of
6675 correspondence between the security properties and the security features shall be formal.
- 6676 **11.4.4.9** The validity of the security properties shall mean that the TOE is in a secure state. By
6677 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
6678 state.
- 6679 **11.4.4.9 Work unit ADV_SPM.1-6**
- 6680 **11.4.4.9** The evaluator *shall examine the TOE security policy model rationale to determine that*
6681 *it proves the internal consistency of the TOE security policy model.*
- 6682 **11.4.4.9** The proof shall show the absence of contradictions within the TOE security policy
6683 model. In determining the absence of contradictions, the evaluator verifies that the arguments
6684 used in the proof are valid.
- 6685 **11.4.4.9** Since the TOE security policy model is formal, the proof of its internal consistency shall
6686 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
6687 security policy model usually is not possible due to the fundamental nature of formal frameworks.
6688 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
6689 security policy model that prove the internal consistency by means of a combination with generic
6690 arguments of the formal framework.
- 6691 **11.4.4.9 ADV_SPM.1.3C** *The correspondence between the model and the functional*
6692 *specification shall be at the correct level of formality.*
- 6693 **11.4.4.9 Work unit ADV_SPM.1-7**
- 6694 **11.4.4.9** The evaluator *shall examine the correspondence between the model and the functional*
6695 *specification to determine that a semiformal demonstration of correspondence between the*
6696 *model and any semiformal functional specification is provided.*
- 6697 **11.4.4.9** This work unit is only applicable to a semiformal presentation of the functional
6698 specification, which is required by ADV_FSP.5.2C.
- 6699 **11.4.4.9** A semiformal correspondence is one that results from a structured approach with a
6700 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
6701 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
6702 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 6703 **11.4.4.9** For guidance on semiformal methods refer to Annex 3.1.1 'Semiformal and formal
6704 methods'.
- 6705 **11.4.4.9 Work unit ADV_SPM.1-8**
- 6706 **11.4.4.9** The evaluator *shall examine the correspondence between the model and the functional*
6707 *specification to determine that a formal proof of correspondence between the model and any*
6708 *formal functional specification is provided.*
- 6709 **11.4.4.9** This work unit is only applicable to a formal presentation of the functional specification,
6710 which is required by ADV_FSP.6.2D.

- 6711 **11.4.4.9** There should be a formal proof of correspondence between the model and any formal
6712 functional specification.
- 6713 **11.4.4.9** The formal proof of correspondence removes all subjective interpretations of its terms
6714 by enlisting well-established mathematical concepts to define the syntax and semantics of the
6715 formal notation and uses rules that support logical reasoning. The security features within the
6716 TOE (which are identified in the formal TSP model) are expressed in a formal specification
6717 language and shown to be satisfied by the formal specification.
- 6718 **11.4.4.9** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 6719 **11.4.4.9 ADV_SPM.1.4C** *The correspondence shall show that the functional*
6720 *specification is consistent and complete with respect to the model.*
- 6721 **11.4.4.9 Work unit ADV_SPM.1-9**
- 6722 **11.4.4.9** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
6723 ***TSF interfaces (as articulated in the functional specification) is complete with respect to the***
6724 ***behaviour modelled by the security features.***
- 6725 **11.4.4.9** The term “correspondence” here means both the formal proof of correspondence
6726 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
6727 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 6728 **11.4.4.9** In determining completeness of the correspondence, the evaluator considers the
6729 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
6730 features of the TSP model. The demonstration should show that all characteristics belonging to
6731 policies that are required to be modelled have an associated feature description in the TOE
6732 security policy model, and that each feature of the TSP model does occur in the mapping.
- 6733 **11.4.4.9** Abstention from formally modelling TSFI behaviour always calls for justification on the
6734 developer’s side (also confer the application notes above).
- 6735 **11.4.4.9 Work unit ADV_SPM.1-10**
- 6736 **11.4.4.9** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
6737 ***TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
6738 ***behaviour modelled by the security features.***
- 6739 **11.4.4.9** The term “correspondence” here means both the formal proof of correspondence
6740 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
6741 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 6742 **11.4.4.9** The meaning of consistency reflects the conventional understanding in contrast to the
6743 internal consistency concept of work unit ADV_SPM.1-6.
- 6744 **11.4.4.9** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
6745 security features established in the preceding work unit and verifies that the correspondence
6746 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
6747 behaviour.
- 6748 **11.4.4.9** For example, if TSFI behaviour dealt with access management on the granularity of
6749 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
6750 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
6751 management for groups of users, then a TSP model describing the security behaviour of the TOE
6752 in terms of individual users would also not be consistent.

6753 **11.4.4.9** As another example, if remote untrusted users had to pass more stringent
 6754 authentication procedures than administrators whose only point of access were within a
 6755 physically-protected area, then this difference in authentication procedures had to be reflected in
 6756 the security features.

6757 **11.4.4.9** TOE design (ADV_TDS)) related that TSFI to determine if the description is complete
 6758 and accurate.

6759 The evaluator determines that, for each TSFI, the exact set of error messages that can be returned
 6760 on invoking that interface can be determined. The evaluator reviews the evidence provided for the
 6761 interface to determine if the set of errors seems complete. They cross-check this information with
 6762 other evidence provided for the evaluation (e.g., TOE design, security architecture description,
 6763 operational user guidance, implementation representation) to ensure that there are no errors
 6764 steaming from processing mentioned that are not included in the functional specification.

6765 **11.4.4.12.11 Work unit ADV_FSP.4-9**

6766 The evaluator **shall examine** the presentation of the TSFI to determine that it completely and
 6767 accurately describes the meaning of all error messages resulting from an invocation of each TSFI.

6768 In order to determine accuracy, the evaluator must be able to understand meaning of the error. For
 6769 example, if an interface returns a numeric code of 0, 1, or 2, the evaluator would not be able to
 6770 understand the error if the functional specification only listed: "possible errors resulting from
 6771 invocation of the *foo()* interface are 0, 1, or 2". Instead the evaluator checks to ensure that the
 6772 errors are described such as: "possible errors resulting from invocation of the *foo()* interface are 0
 6773 (processing successful), 1 (file not found), or 2 (incorrect filename specification)".

6774 In order to determine that the description of the errors due to invoking a TSFI is complete, the
 6775 evaluator examines the rest of the interface description (parameter descriptions, actions, etc.) to
 6776 determine if potential error conditions that might be caused by using such an interface are
 6777 accounted for. The evaluator also checks other evidence provided for the evaluation (e.g. TOE
 6778 design, security architecture description, operational user guidance, implementation
 6779 representation) to see if error processing related to the TSFI is described there but is not described
 6780 in the functional specification.

6781 ISO/IEC 15408-3 ADV_FSP.4.6C: *The tracing shall demonstrate that the SFRs trace to TSFIs in the*
 6782 *functional specification.*

6783 **11.4.4.12.12 Work unit ADV_FSP.4-10**

6784 The evaluator **shall check** that the tracing links the SFRs to the corresponding TSFIs.

6785 The tracing is provided by the developer to serve as a guide to which SFRs are related to which
 6786 TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the
 6787 following work units, in which the evaluator verifies its completeness and accuracy.

6788 **11.4.4.13 Action ADV_FSP.4.2E**

6789 **11.4.4.13.1 Work unit ADV_FSP.4-11**

6790 The evaluator **shall examine** the functional specification to determine that it is a complete
 6791 instantiation of the SFRs.

6792 To ensure that all SFRs are covered by the functional specification, as well as the test coverage
 6793 analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.4-10 a map between
 6794 the TOE security functional requirements and the TSFI. Note that this map may have to be at a level
 6795 of detail below the component or even element level of the requirements, because of operations
 6796 (assignments, refinements, selections) performed on the functional requirement by the ST author.

6797 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained,
 6798 for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three
 6799 different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and
 6800 claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to
 6801 TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper
 6802 interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of
 6803 parameters for a given interface.

6804 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 6805 boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the
 6806 TSFI. The analysis for those requirements will be performed in the analysis for the TOE design
 6807 (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions,
 6808 and error messages associated with TSFIs must be fully specified, the evaluator should be able to
 6809 determine if all aspects of an SFR appear to be implemented at the interface level.

6810 **11.4.4.13.2 Work unit ADV_FSP.4-12**

6811 The evaluator *shall examine* the functional specification to determine that it is an accurate
 6812 instantiation of the SFRs.

6813 For each functional requirement in the ST that results in effects visible at the TSF boundary, the
 6814 information in the associated TSFI for that requirement specifies the required functionality
 6815 described by the requirement. For example, if the ST contains a requirement for access control lists,
 6816 and the only TSFI that map to that requirement specify functionality for Unix-style protection bits,
 6817 then the functional specification is not accurate with respect to the requirements.

6818 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 6819 boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements
 6820 to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE
 6821 design (ADV_TDS) when included in the ST.

6822 **11.4.5 Evaluation of sub-activity (ADV_FSP.5)**

6823 **11.4.5.1 Objectives**

6824 The objective of this sub-activity is to determine whether the developer has completely described
 6825 all of the TSFI in a manner such that the evaluator is able to determine whether the TSFI are
 6826 completely and accurately described, and appears to implement the security functional
 6827 requirements of the ST. The completeness of the interfaces is judged based upon the
 6828 implementation representation.

6829 **11.4.5.2 Input**

6830 The evaluation evidence for this sub-activity that is required by the work-units is:

- 6831 a) the ST;
- 6832 b) the functional specification;
- 6833 c) the TOE design;
- 6834 d) the implementation representation.

6835 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

- 6836 a) the security architecture description;
- 6837 b) the TSF internals description;

6838 c) the formal security policy model;

6839 d) the operational user guidance;

6840 **11.4.5.3 Action ADV_FSP.5.1E**

6841 ISO/IEC 15408-3 ADV_FSP.5.1C: *The functional specification shall completely represent the TSF.*

6842 **11.4.5.3.1 Work unit ADV_FSP.5-1**

6843 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully
6844 represented.

6845 **11.4.5.4 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity.** 6846 **The TSF must be identified (done as part of the Objectives)**

6847 **11.4.5.4** The objectives of this sub-activity are to determine whether the formal security policy model of
6848 the TSF clearly and consistently describes the rules and characteristics of the security policies
6849 and whether this description corresponds with the description of security functions in the
6850 functional specification.

6851 **11.4.5.4 Input**

6852 **11.4.5.4** The evaluation evidence for this sub-activity is:

6853 **11.4.5.4** the ST;

6854 **11.4.5.4** the functional specification;

6855 **11.4.5.4** formal security policy model (ADV_SPM.1.1D);

6856 **11.4.5.4** formal proof of correspondence between the model and any formal functional specification
6857 (ADV_SPM.1.3D);

6858 **11.4.5.4** demonstration of correspondence between the model and the functional specification
6859 (ADV_SPM.1.4D).

6860 **11.4.5.4 Application notes**

6861 **11.4.5.4** This activity applies to cases where the developer has provided a formal security policy model of
6862 the TOE.

6863 **11.4.5.4** A formal TOE security policy model is a representation of the rules (synonymously termed
6864 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
6865 Their formal counterparts are called security properties and security features, respectively. The
6866 representation includes but is not limited to algebraic specifications, finite state machines and
6867 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
6868 model is accompanied by an informal interpretation explaining how the rules and characteristics
6869 are mapped to the respective properties and features.

6870 **11.4.5.4** The creation of a formal security policy model helps to identify and eliminate ambiguous,
6871 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
6872 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
6873 of how well the developer has understood the security functionality being implemented and
6874 whether there are inconsistencies between the security requirements and the TOE design. The
6875 confidence in the model is accompanied by a proof that it contains no inconsistencies.

- 6876 **11.4.5.4** A formal security model is a precise formal presentation of the important aspects of
6877 security and their relationship to the behaviour of the TOE; it identifies the set of rules
6878 (principles) that defines the TOE security policy and the set of practises (characteristics) that
6879 regulates how the TSF manages, protects, and otherwise controls the system resources. The
6880 model includes the set of restrictions and properties that specify how information and computing
6881 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
6882 engineering arguments showing that these restrictions and properties play a key role in the
6883 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
6884 as well as ancillary text to explain the model and to provide it with context. The security
6885 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
6886 with the rest of the TOE and with its operational environment), as well as its internal behaviour.
- 6887 **11.4.5.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
6888 considering the proposed security requirements of the ST. The informal abstraction is taken to be
6889 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
6890 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
6891 are always prone to fallacies; especially if relationships among subjects, objects and operations
6892 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
6893 characteristics of the security policy model are mapped to respective properties and features
6894 within some formal system, whose rigour and strength can afterwards be used to obtain the
6895 security properties by means of theorems and formal proof.
- 6896 **11.4.5.4** While the term "formal security policy model" is used in academic circles, the CC's
6897 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
6898 claimed. Therefore, the formal security policy model is merely a formal representation of the set
6899 of SFRs being claimed.
- 6900 **11.4.5.4** The term security policy has traditionally been associated with only access control
6901 policies, whether label-based (mandatory access control) or user-based (discretionary access
6902 control). However, a security policy is not limited to access control; there are also audit policies,
6903 identification policies, authentication policies, encryption policies, management policies, and any
6904 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
6905 contains an assignment for identifying these policies that are formally modelled.
- 6906 **11.4.5.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
6907 because either a given policy can not be formally modelled in the otherwise well suited
6908 framework, or because the nature of the TOE renders impossible the modelling of policies that
6909 would otherwise be possible to model.
- 6910 **11.4.5.4 Action ADV_SPM.1.1E**
- 6911 **11.4.5.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
6912 *text as required, and identify the security policies of the TSF that are modelled.*
- 6913 **11.4.5.4 Work unit ADV_SPM.1-1**
- 6914 **11.4.5.4** The evaluator ***shall examine*** the TOE security policy model to determine that it is
6915 written in a formal style.
- 6916 **11.4.5.4** The evaluator identifies the formal framework upon which the TOE security policy
6917 model is based and ensures that it is founded on well established mathematical concepts. **They**
6918 **also identify the security properties and features addressed in the application notes and ensure**
6919 **the formalization of at least one security policy.**
- 6920 **11.4.5.4** For guidance on formal methods refer to ISO/IEC 15408-3

- 6921 **11.4.5.4 Work unit ADV_SPM.1-2**
- 6922 **11.4.5.4** The evaluator *shall examine the TOE security policy model to determine that it*
 6923 contains all necessary informal explanatory text.
- 6924 **11.4.5.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
 6925 to make clear the meaning of any formal notation and how they are used) including the security
 6926 properties and features.
- 6927 **11.4.5.4 Work unit ADV_SPM.1-3**
- 6928 **11.4.5.4** The evaluator *shall examine the TOE security policy model to determine that all*
 6929 security policies of the TSF are identified that are modelled.
- 6930 **11.4.5.4** The evaluator determines whether the SPM identifies the security policies for which a
 6931 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 6932 of the modelled policies.
- 6933 **11.4.5.4** The evaluator determines whether the list of security policies identified by the SPM is
 6934 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 6935 **11.4.5.4** The evaluator determines whether for each security policy identified by the SPM a
 6936 model is in fact provided.
- 6937 **11.4.5.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 6938 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 6939 *not secure.*
- 6940 **11.4.5.4 Work unit ADV_SPM.1-4**
- 6941 **11.4.5.4** The evaluator *shall examine the principles and characteristics of the security policies*
 6942 to determine that the modelled security behaviour of the TOE is clearly articulated.
- 6943 **11.4.5.4** The security policies are expressed in terms of security principles (rules) which are
 6944 modelled by security properties and define the secure state of the TOE. For example, a model
 6945 based on state transitions could describe the security policies in terms of principles of its states,
 6946 identify its initial state, and define what it means to be a secure state.
- 6947 **11.4.5.4** The evaluator determines that the security policies are reflected within their formal
 6948 counterparts of the TSP model.
- 6949 **11.4.5.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 6950 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 6951 resources including attributes and conditions of the TOE) which are modelled by security
 6952 features. For example, a model based on state transitions could describe the characteristics as
 6953 possible actions in each secure state in a level of detail sufficient to decide into which state the
 6954 TOE will be transformed by that action.
- 6955 **11.4.5.4** Together the security principles and characteristics describe the entire security posture
 6956 of the TOE.
- 6957 **11.4.5.4** In the context of a formal TOE security policy model the security behaviour is
 6958 considered to be clearly articulated only if an adequate mapping from principles and
 6959 characteristics to their respective formal counterparts properties and features has been given.
 6960 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 6961 detailed enough to allow for correct identification of all security objectives and the relation to the
 6962 security environment.

- 6963 **11.4.5.4** The above condition for clear articulation is necessary but not sufficient. An informal
6964 interpretation of all formal concepts (including attributes, predicates and variables, if available)
6965 must be provided in order to make clear their intended meaning.
- 6966 **11.4.5.4 Work unit ADV_SPM.1-5**
- 6967 **11.4.5.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
6968 it formally proves that the security features enforce the security properties.
- 6969 **11.4.5.4** To determine the enforcement, the evaluator considers the security properties and the
6970 security features and verifies that the arguments used in the proof are valid. The proof of
6971 correspondence between the security properties and the security features shall be formal.
- 6972 **11.4.5.4** The validity of the security properties shall mean that the TOE is in a secure state. By
6973 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
6974 state.
- 6975 **11.4.5.4 Work unit ADV_SPM.1-6**
- 6976 **11.4.5.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
6977 it proves the internal consistency of the TOE security policy model.
- 6978 **11.4.5.4** The proof shall show the absence of contradictions within the TOE security policy
6979 model. In determining the absence of contradictions, the evaluator verifies that the arguments
6980 used in the proof are valid.
- 6981 **11.4.5.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
6982 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
6983 security policy model usually is not possible due to the fundamental nature of formal frameworks.
6984 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
6985 security policy model that prove the internal consistency by means of a combination with generic
6986 arguments of the formal framework.
- 6987 **11.4.5.4 ADV_SPM.1.3C** *The correspondence between the model and the functional*
6988 *specification shall be at the correct level of formality.*
- 6989 **11.4.5.4 Work unit ADV_SPM.1-7**
- 6990 **11.4.5.4** The evaluator *shall examine the correspondence between the model and the functional*
6991 *specification to determine that a semiformal demonstration of correspondence between the*
6992 *model and any semiformal functional specification is provided.*
- 6993 **11.4.5.4** This work unit is only applicable to a semiformal presentation of the functional
6994 specification, which is required by ADV_FSP.5.2C.
- 6995 **11.4.5.4** A semiformal correspondence is one that results from a structured approach with a
6996 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
6997 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
6998 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 6999 **11.4.5.4** For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
7000 methods’.
- 7001 **11.4.5.4 Work unit ADV_SPM.1-8**
- 7002 **11.4.5.4** The evaluator *shall examine the correspondence between the model and the functional*
7003 *specification to determine that a formal proof of correspondence between the model and any*
7004 *formal functional specification is provided.*

- 7005 **11.4.5.4** This work unit is only applicable to a formal presentation of the functional specification,
7006 which is required by ADV_FSP.6.2D.
- 7007 **11.4.5.4** There should be a formal proof of correspondence between the model and any formal
7008 functional specification.
- 7009 **11.4.5.4** The formal proof of correspondence removes all subjective interpretations of its terms
7010 by enlisting well-established mathematical concepts to define the syntax and semantics of the
7011 formal notation and uses rules that support logical reasoning. The security features within the
7012 TOE (which are identified in the formal TSP model) are expressed in a formal specification
7013 language and shown to be satisfied by the formal specification.
- 7014 **11.4.5.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 7015 **11.4.5.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
7016 *specification is consistent and complete with respect to the model.*
- 7017 **11.4.5.4 Work unit ADV_SPM.1-9**
- 7018 **11.4.5.4** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
7019 ***TSF interfaces (as articulated in the functional specification) is complete with respect to the***
7020 ***behaviour modelled by the security features.***
- 7021 **11.4.5.4** The term “correspondence” here means both the formal proof of correspondence
7022 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
7023 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 7024 **11.4.5.4** In determining completeness of the correspondence, the evaluator considers the
7025 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
7026 features of the TSP model. The demonstration should show that all characteristics belonging to
7027 policies that are required to be modelled have an associated feature description in the TOE
7028 security policy model, and that each feature of the TSP model does occur in the mapping.
- 7029 **11.4.5.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
7030 developer’s side (also confer the application notes above).
- 7031 **11.4.5.4 Work unit ADV_SPM.1-10**
- 7032 **11.4.5.4** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
7033 ***TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
7034 ***behaviour modelled by the security features.***
- 7035 **11.4.5.4** The term “correspondence” here means both the formal proof of correspondence
7036 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
7037 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 7038 **11.4.5.4** The meaning of consistency reflects the conventional understanding in contrast to the
7039 internal consistency concept of work unit ADV_SPM.1-6.
- 7040 **11.4.5.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
7041 security features established in the preceding work unit and verifies that the correspondence
7042 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
7043 behaviour.
- 7044 **11.4.5.4** For example, if TSFI behaviour dealt with access management on the granularity of
7045 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
7046 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access

- 7047 management for groups of users, then a TSP model describing the security behaviour of the TOE
7048 in terms of individual users would also not be consistent.
- 7049 **11.4.5.4** As another example, if remote untrusted users had to pass more stringent
7050 authentication procedures than administrators whose only point of access were within a
7051 physically-protected area, then this difference in authentication procedures had to be reflected in
7052 the security features.
- 7053 **11.4.5.4** TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be
7054 done at a high level to ensure that no large groups of interfaces have been missed (network
7055 protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the
7056 functional specification proceeds.
- 7057 In making an assessment for this work unit, the evaluator determines that all portions of the TSF
7058 are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF
7059 should have a corresponding interface description, or if there are no corresponding interfaces for a
7060 portion of the TSF, the evaluator determines that that is acceptable.
- 7061 ISO/IEC 15408-3 ADV_FSP.5.2C: *The functional specification shall describe the TSFI using a semi-*
7062 *formal style.*
- 7063 **11.4.5.7.11 Work unit ADV_FSP.5-2**
- 7064 The evaluator ***shall examine*** the functional specification to determine that it is presented using a
7065 semiformal style.
- 7066 A semi-formal presentation is characterised by a standardised format with a well-defined syntax
7067 that reduces ambiguity that may occur in informal presentations. Since the intent of the semi-
7068 formal format is to enhance the reader's ability to understand the presentation, use of certain
7069 structured presentation methods (pseudo-code, flow charts, block diagrams) are appropriate,
7070 though not required.
- 7071 For the purposes of this activity, the evaluator should ensure that the interface descriptions are
7072 formatted in a structured, consistent manner and use common terminology. A semiformal
7073 presentation of the interfaces also implies that the level of detail of the presentation for the
7074 interfaces is largely consistent across all TSFI. For the functional specification, it is acceptable to
7075 refer to external specifications for portions of the interface as long as those external specifications
7076 are themselves semiformal.
- 7077 ISO/IEC 15408-3 ADV_FSP.5.3C: *The functional specification shall describe the purpose and method*
7078 *of use for all TSFI.*
- 7079 **11.4.5.7.12 Work unit ADV_FSP.5-3**
- 7080 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of
7081 each TSFI.
- 7082 The purpose of a TSFI is a general statement summarising the functionality provided by the
7083 interface. It is not intended to be a complete statement of the actions and results related to the
7084 interface, but rather a statement to help the reader understand in general what the interface is
7085 intended to be used for. The evaluator should not only determine that the purpose exists, but also
7086 that it accurately reflects the TSFI by taking into account other information about the interface,
7087 such as the description of actions and error messages.
- 7088 Work unit ADV_FSP.5-4
- 7089 The evaluator ***shall examine*** the functional specification to determine that the method of use for
7090 each TSFI is given.

- 7091 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the
 7092 actions and obtain the results associated with the TSFI. The evaluator should be able to determine,
 7093 from reading this material in the functional specification, how to use each interface. This does not
 7094 necessarily mean that there needs to be a separate method of use for each TSFI, as it may be
 7095 possible to describe in general how kernel calls are invoked, for instance, and then identify each
 7096 interface using that general style. Different types of interfaces will require different method of use
 7097 specifications. APIs, network protocol interfaces, system configuration parameters, and hardware
 7098 bus interfaces all have very different methods of use, and this should be taken into account by the
 7099 developer when developing the functional specification, as well as by the evaluator evaluating the
 7100 functional specification.
- 7101 For administrative interfaces whose functionality is documented as being inaccessible to untrusted
 7102 users, the evaluator ensures that the method of making the functions inaccessible is described in
 7103 the functional specification. It should be noted that this inaccessibility needs to be tested by the
 7104 developer in their test suite.
- 7105 The evaluator should not only determine that the set of method of use descriptions exist, but also
 7106 that they accurately cover each TSFI.
- 7107 **11.4.5.7.13 Work unit ADV_FSP.5-5**
- 7108 The evaluator *shall examine* the functional specification to determine the completeness of the
 7109 TSFI
- 7110 The evaluator shall use the design documentation to identify the possible types of interfaces. The
 7111 evaluator shall search the design documentation and the guidance documentation for potential
 7112 TSFI not contained in the developer's documentation, thus indicating that the set of TSFI defined
 7113 by the developer is incomplete. The evaluator *shall examine* the arguments presented by the
 7114 developer that the TSFI is complete and check down to the lowest level of design or with the
 7115 implementation representation that no additional TSFI exist.
- 7116 ISO/IEC 15408-3 ADV_FSP.5.4C: *The functional specification shall identify and describe all*
 7117 *parameters associated with each TSFI.*
- 7118 **11.4.5.7.14 Work unit ADV_FSP.5-6**
- 7119 The evaluator *shall examine* the presentation of the TSFI to determine that it completely identifies
 7120 all parameters associated with every TSFI.
- 7121 The evaluator examines the functional specification to ensure that all of the parameters are
 7122 described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the
 7123 behaviour of that interface. For examples, parameters are the arguments supplied to an API; the
 7124 various fields in packet for a given network protocol; the individual key values in the Windows
 7125 Registry; the signals across a set of pins on a chip; etc.
- 7126 In order to determine that all of the parameters are present in the TSFI, the evaluator should
 7127 examine the rest of the interface description (actions, error messages, etc.) to determine if the
 7128 effects of the parameter are accounted for in the description. The evaluator should also check other
 7129 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 7130 operational user guidance, implementation representation) to see if behaviour or additional
 7131 parameters are described there but not in the functional specification.
- 7132 **11.4.5.7.15 Work unit ADV_FSP.5-7**
- 7133 The evaluator *shall examine* the presentation of the TSFI to determine that it completely and
 7134 accurately describes all parameters associated with every TSFI.

7135 Once all of the parameters have been identified, the evaluator needs to ensure that they are
 7136 accurately described, and that the description of the parameters is complete. A parameter
 7137 description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)*
 7138 could be described as having “parameter *i* which is an integer”; this is not an acceptable parameter
 7139 description. A description such as “parameter *i* is an integer that indicates the number of users
 7140 currently logged in to the system”. is much more acceptable.

7141 In order to determine that the description of the parameters is complete, the evaluator should
 7142 examine the rest of the interface description (purpose, method of use, actions, error messages, etc.)
 7143 to determine if the descriptions of the parameter(s) are accounted for in the description. The
 7144 evaluator should also check other evidence provided (e.g., TOE design, architectural design,
 7145 operational user guidance, implementation representation) to see if behaviour or additional
 7146 parameters are described there but not in the functional specification.

7147 ISO/IEC 15408-3 ADV_FSP.5.5C: *The functional specification shall describe all actions associated*
 7148 *with each TSFI.*

7149 **11.4.5.7.16 Work unit ADV_FSP.5-8**

7150 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 7151 accurately describes all actions associated with every TSFI.

7152 The evaluator checks to ensure that all of the actions are described. actions available through an
 7153 interface describe what the interface does (as opposed to the TOE design, which describes how the
 7154 actions are provided by the TSF).

7155 actions of an interface describe functionality that can be invoked through the interface, and can be
 7156 categorised as *regular* actions, and *SFR-related* actions. Regular actions are descriptions of what the
 7157 interface does. The amount of information provided for this description is dependant on the
 7158 complexity of the interface. The SFR-related actions are those that are visible at any external
 7159 interface (for instance, audit activity caused by the invocation of an interface (assuming audit
 7160 requirements are included in the ST) should be described, even though the result of that action is
 7161 generally not visible through the invoked interface). Depending on the parameters of an interface,
 7162 there may be many different actions able to be invoked through the interface (for instance, an API
 7163 might have the first parameter be a “subcommand”, and the following parameters be specific to
 7164 that subcommand. The IOCTL API in some Unix systems is an example of such an interface).

7165 In order to determine that the description of the actions of a TSFI is complete, the evaluator should
 7166 review the rest of the interface description (parameter descriptions, error messages, etc.) to
 7167 determine if the actions described are accounted for. The evaluator should also analyse other
 7168 evidence provided for the evaluation (e.g., TOE design, security architecture description,
 7169 operational user guidance, implementation representation) to see if there is evidence of actions
 7170 that are described there but not in the functional specification.

7171 ISO/IEC 15408-3 ADV_FSP.5.6C: *The functional specification shall describe all direct error messages*
 7172 *that may result from an invocation of each TSFI.*

7173 **11.4.5.7.17 Work unit ADV_FSP.5-9**

7174 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 7175 accurately describes all errors messages resulting from an invocation of each TSFI.

7176 Errors can take many forms, depending on the interface being described. For an API, the interface
 7177 itself may return an error code; set a global error condition, or set a certain parameter with an
 7178 error code. For a configuration file, an incorrectly configured parameter may cause an error
 7179 message to be written to a log file. For a hardware PCI card, an error condition may raise a signal
 7180 on the bus, or trigger an exception condition to the CPU.

7181 Errors (and the associated error messages) come about through the invocation of an interface. The
 7182 processing that occurs in response to the interface invocation may encounter error conditions,
 7183 which trigger (through an implementation-specific mechanism) an error message to be generated.
 7184 In some instances this may be a return value from the interface itself; in other instances a global
 7185 value may be set and checked after the invocation of an interface. It is likely that a TOE will have a
 7186 number of low-level error messages that may result from fundamental resource conditions, such as
 7187 “disk full” or “resource locked”. While these error messages may map to a large number of TSFI,
 7188 they could be used to detect instances where detail from an interface description has been omitted.
 7189 For instance, a TSFI that produces a “disk full” message, but has no obvious description of why that
 7190 TSFI should cause an access to the disk in its description of actions, might cause the evaluator to
 7191 examine other evidence (ADV_ARC, ADV_TDS) related that TSFI to determine if the description is
 7192 complete and accurate.

7193 The evaluator determines that, for each TSFI, the exact set of error messages that can be returned
 7194 on invoking that interface can be determined. The evaluator reviews the evidence provided for the
 7195 interface to determine if the set of errors seems complete. They cross-check this information with
 7196 other evidence provided for the evaluation (e.g., TOE design, security architecture description,
 7197 operational user guidance, implementation representation) to ensure that there are no errors
 7198 steaming from processing mentioned that are not included in the functional specification.

7199 **11.4.5.7.18 Work unit ADV_FSP.5-10**

7200 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and
 7201 accurately describes the meaning of all error messages resulting from an invocation of each TSFI.

7202 In order to determine accuracy, the evaluator must be able to understand meaning of the error. For
 7203 example, if an interface returns a numeric code of 0, 1, or 2, the evaluator would not be able to
 7204 understand the error if the functional specification only listed: “possible errors resulting from
 7205 invocation of the *foo()* interface are 0, 1, or 2”. Instead the evaluator checks to ensure that the
 7206 errors are described such as: “possible errors resulting from invocation of the *foo()* interface are 0
 7207 (processing successful), 1 (file not found), or 2 (incorrect filename specification)”.

7208 In order to determine that the description of the errors due to invoking a TSFI is complete, the
 7209 evaluator examines the rest of the interface description (parameter descriptions, actions, etc.) to
 7210 determine if potential error conditions that might be caused by using such an interface are
 7211 accounted for. The evaluator also checks other evidence provided for the evaluation (e.g., TOE
 7212 design, security architecture description, operational user guidance, implementation
 7213 representation) to see if error processing related to the TSFI is described there but is not described
 7214 in the functional specification.

7215 ISO/IEC 15408-3 ADV_FSP.5.7C: *The functional specification shall describe all error messages that*
 7216 *do not result from an invocation of a TSFI.*

7217 **11.4.5.7.19 Work unit ADV_FSP.5-11**

7218 The evaluator ***shall examine*** the functional specification to determine that it completely and
 7219 accurately describes all errors messages that do not result from an invocation of any TSFI.

7220 This work unit complements work unit ADV_FSP.5-9, which describes those error messages that
 7221 result from an invocation of the TSFI. Taken together, these work units cover all error messages
 7222 that might be generated by the TSF.

7223 The evaluator assesses the completeness and accuracy of the functional specification by comparing
 7224 its contents to instances of error message generation within the implementation representation.
 7225 Most of these error messages will have already been covered by work unit ADV_FSP.5-9.

7226 The error messages related to this work unit are typically those that are not expected to be
 7227 generated, but are constructed as a matter of good programming practises. For example, a case

7228 statement that defines actions resulting from each of a list of cases may end with a final *else*
 7229 statement to apply to anything that might not be expected; this practise ensures the TSF does not
 7230 get into an undefined state. However, it is not expected that the path of execution would ever get to
 7231 this *else* statement; therefore, any error message generation within this *else* statement would never
 7232 be generated. Although it would not get generated, it must still be included in the functional
 7233 specification.

7234 ISO/IEC 15408-3 ADV_FSP.5.8C: *The functional specification shall provide a rationale for each error*
 7235 *message contained in the TSF implementation yet does not result from an invocation of a TSFI.*

7236 **11.4.5.7.20 Work unit ADV_FSP.5-12**

7237 The evaluator ***shall examine*** the functional specification to determine that it provides a rationale
 7238 for each error message contained in the TSF implementation yet does not result from an invocation
 7239 of a TSFI.

7240 The evaluator ensures that every error message found under work unit ADV_FSP.5-11 contains a
 7241 rationale describing why it cannot be invoked from the TSFI.

7242 As was described in the previous work unit, this rationale might be as straightforward as the fact
 7243 that the error message in question is provided for completeness of execution logic and that it is
 7244 never expected to be generated. The evaluator ensures that the rationale for each such error
 7245 message is logical.

7246 ISO/IEC 15408-3 ADV_FSP.5.9C: *The tracing shall demonstrate that the SFRs trace to TSFIs in the*
 7247 *functional specification.*

7248 **11.4.5.7.21 Work unit ADV_FSP.5-13**

7249 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

7250 The tracing is provided by the developer to serve as a guide to which SFRs are related to which
 7251 TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the
 7252 following work units, in which the evaluator verifies its completeness and accuracy.

7253 **11.4.5.8 Action ADV_FSP.5.2E**

7254 **11.4.5.8.1 Work unit ADV_FSP.5-14**

7255 The evaluator ***shall examine*** the functional specification to determine that it is a complete
 7256 instantiation of the SFRs.

7257 To ensure that all SFRs are covered by the functional specification, as well as the test coverage
 7258 analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.5-13 a map between
 7259 the TOE security functional requirements and the TSFI. Note that this map may have to be at a level
 7260 of detail below the component or even element level of the requirements, because of operations
 7261 (assignments, refinements, selections) performed on the functional requirement by the ST author.

7262 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained,
 7263 for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three
 7264 different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and
 7265 claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to
 7266 TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper
 7267 interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of
 7268 parameters for a given interface.

7269 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 7270 boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the

7271 TSFI. The analysis for those requirements will be performed in the analysis for the TOE design
 7272 (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions,
 7273 and error messages associated with TSFIs must be fully specified, the evaluator should be able to
 7274 determine if all aspects of an SFR appear to be implemented at the interface level.

7275 **11.4.5.8.2 Work unit ADV_FSP.5-15**

7276 The evaluator ***shall examine*** the functional specification to determine that it is an accurate
 7277 instantiation of the SFRs.

7278 For each functional requirement in the ST that results in effects visible at the TSF boundary, the
 7279 information in the associated TSFI for that requirement specifies the required functionality
 7280 described by the requirement. For example, if the ST contains a requirement for access control lists,
 7281 and the only TSFI that map to that requirement specify functionality for Unix-style protection bits,
 7282 then the functional specification is not accurate with respect to the requirements.

7283 The evaluator must recognise that for requirements that have little or no manifestation at the TSF
 7284 boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements
 7285 to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE
 7286 design (ADV_TDS) when included in the ST.

7287 **11.4.6 Evaluation of sub-activity (ADV_FSP.6)**

7288 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

7289 **11.5 Implementation representation (ADV_IMP)**

7290 **11.5.1 Evaluation of sub-activity (ADV_IMP.1)**

7291 **11.5.1.1 Objectives**

7292 The objective of this sub-activity is to determine that the implementation representation made
 7293 available by the developer is suitable for use in other analysis activities; *suitability* is judged by its
 7294 conformance to the requirements for this component.

7295 **11.5.1.2 Input**

7296 The evaluation evidence for this sub-activity is:

- 7297 a) the implementation representation;
- 7298 b) the documentation of the development tools, as resulting from ALC_TAT ;
- 7299 c) TOE design description.

7300 **11.5.1.3 Application notes**

7301 The entire implementation representation is made available to ensure that analysis activities are
 7302 not curtailed due to lack of information. This does not, however, imply that all of the representation
 7303 is examined when the analysis activities are being performed. This is likely impractical in almost all
 7304 cases, in addition to the fact that it most likely will not result in a higher-assurance TOE vs. targeted
 7305 sampling of the implementation representation. For this sub-activity, this is even truer. It would
 7306 not be productive for the evaluator to spend large amounts of time verifying the requirements for
 7307 one portion of the implementation representation, and then use a different portion of the
 7308 implementation representation in performing analysis for other work units. Therefore, the
 7309 evaluator is encouraged to select the sample of the implementation representation from the areas
 7310 of the TOE that will be of most interest during the analysis performed during work units from other
 7311 families (e.g. ATE_IND, AVA_VAN and ADV_INT).

7312 **11.5.1.4 Action ADV_IMP.1.1E**

7313 ISO/IEC 15408-3 ADV_IMP.1.1C: *The implementation representation shall define the TSF to a level of*
 7314 *detail such that the TSF can be generated without further design decisions.*

7315 **11.5.1.4.1 Work unit ADV_IMP.1-1**

7316 The evaluator ***shall check*** that the implementation representation defines the TSF to a level of
 7317 detail such that the TSF can be generated without further design decisions.

7318 Source code or hardware diagrams and/or IC hardware design language code or layout data that
 7319 are used to build the actual hardware are examples of parts of an implementation representation.
 7320 The evaluator samples the implementation representation to gain confidence that it is at the
 7321 appropriate level and not, for instance, a pseudo-code level which requires additional design
 7322 decisions to be made. The evaluator is encouraged to perform a quick check when first looking at
 7323 the implementation representation to assure themselves that the developer is on the right track.
 7324 However, the evaluator is also encourage to perform the bulk of this check while working on other
 7325 work units that call for examining the implementation; this will ensure the sample examined for
 7326 this work unit is relevant.

7327 ISO/IEC 15408-3 ADV_IMP.1.2C: *The implementation representation shall be in the form used by the*
 7328 *development personnel.*

7329 **11.5.1.4.2 Work unit ADV_IMP.1-2**

7330 The evaluator ***shall check*** that the implementation representation is in the form used by
 7331 development personnel.

7332 The implementation representation is manipulated by the developer in form that it suitable for
 7333 transformation to the actual implementation. For instance, the developer may work with files
 7334 containing source code, which is eventually compiled to become part of the TSF. The developer
 7335 makes available the implementation representation in the form they use, so that the evaluator may
 7336 use automated techniques in the analysis. This also increases the confidence that the
 7337 implementation representation examined is actually the one used in the production of the TSF (as
 7338 opposed to the case where it is supplied in an alternate presentation format, such as a word
 7339 processor document). It should be noted that other forms of the implementation representation
 7340 may also be used by the developer; these forms are supplied as well. The overall goal is to supply
 7341 the evaluator with the information that will maximise the evaluator's analysis efforts.

7342 The evaluator samples the implementation representation to gain confidence that it is the version
 7343 that is usable by the developer. The sample is such that the evaluator has assurance that all areas of
 7344 the implementation representation are in conformance with the requirement; however, a complete
 7345 examination of the entire implementation representation is unnecessary.

7346 Conventions in some forms of the implementation representation may make it difficult or
 7347 impossible to determine from just the implementation representation itself what the actual result
 7348 of the compilation or run-time interpretation will be. For example, compiler directives for C
 7349 language compilers will cause the compiler to exclude or include entire portions of the code.

7350 Some forms of the implementation representation may require additional information because
 7351 they introduce significant barriers to understanding and analysis. Examples include shrouded
 7352 source code or source code that has been obfuscated in other ways such that it prevents
 7353 understanding and/or analysis. These forms of implementation representation typically result
 7354 from by taking a version of the implementation representation that is used by the TOE developer
 7355 and running a shrouding or obfuscation program on it. While the shrouded representation is what
 7356 is compiled and may be closer to the implementation (in terms of structure) than the original, un-
 7357 shrouded representation, supplying such obfuscated code may cause significantly more time to be
 7358 spent in analysis tasks involving the representation. When such forms of representation are

7359 created, the components require details on the shrouding tools/algorithms used so that the un-
 7360 shrouded representation can be supplied, and the additional information can be used to gain
 7361 confidence that the shrouding process does not compromise any security mechanisms.

7362 The evaluator samples the implementation representation to gain confidence that all of the
 7363 information needed to interpret the implementation representation has been supplied. Note that
 7364 the tools are among those referenced by Tools and techniques (ALC_TAT) components. The
 7365 evaluator is encouraged to perform a quick check when first looking at the implementation
 7366 representation to assure themselves that the developer is on the right track. However, the
 7367 evaluator is also encouraged to perform the bulk of this check while working on other work units
 7368 that call for examining the implementation; this will ensure the sample examined for this work unit
 7369 is relevant.

7370 ISO/IEC 15408-3 ADV_IMP.1.3C: *The mapping between the TOE design description and the sample of*
 7371 *the implementation representation shall demonstrate their correspondence.*

7372 **11.5.1.4.3 Work unit ADV_IMP.1-3**

7373 The evaluator ***shall examine*** the mapping between the TOE design description and the sample of
 7374 the implementation representation to determine that it is accurate.

7375 The evaluator augments the determination of existence (specified in work unit ADV_IMP.1-1) by
 7376 verifying the accuracy of a portion of the implementation representation and the TOE design
 7377 description. For parts of the TOE design description that are interesting, the evaluator would verify
 7378 the implementation representation accurately reflects the description provided in the TOE design
 7379 description.

7380 For example, the TOE design description might identify a login module that is used to identify and
 7381 authenticate users. If user authentication is sufficiently significant, the evaluator would verify that
 7382 the corresponding code in fact implements that service as described in the TOE design description.
 7383 It might also be worthwhile to verify that the code accepts the parameters as described in the
 7384 functional specification.

7385 It is worth pointing out the developer must choose whether to perform the mapping for the entire
 7386 implementation representation, thereby guaranteeing that the chosen sample will be covered, or
 7387 waiting for the sample to be chosen before performing the mapping. The first option is likely more
 7388 work, but may be completed before the evaluation begins. The second option is less work, but will
 7389 produce a suspension of evaluation activity while the necessary evidence is being produced.

7390 **11.5.2 Evaluation of sub-activity (ADV_IMP.2)**

7391

7392 **11.5.2.1 Objectives**

7393 The objective of this sub-activity is to determine that the implementation representation made
 7394 available by the developer is suitable for use in other analysis activities; suitability is judged by its
 7395 conformance to the requirements for this component.

7396 **11.5.2.2 Input**

7397 The evaluation evidence for this sub-activity is:

7398

7399 d) the implementation representation;

7400 e) the documentation of the development tools, as resulting from ALC_TAT;

7401 f) the TOE design description.

7402 11.5.2.3 Application notes

7403 The entire implementation representation is made available to ensure that analysis activities are
7404 not curtailed due to lack of information. This does not, however, imply that all of the representation
7405 is examined in detail when the analysis activities are being performed. This is likely impractical in
7406 almost all cases, in addition to the fact that it most likely will not result in a higher-assurance TOE.

7407 The new aspect for ADV_IMP.2 in comparison to ADV_IMP.1 is that the developer needs to
7408 demonstrate and the evaluator will confirm that the complete implementation representation is
7409 mapped to the TOE design description. This does, however, not imply that all other work units
7410 need an examination of the complete implementation representation. Aspects like appropriate
7411 level of detail and form of the implementation representation can be covered by sampling as for
7412 ADV_IMP.1.

7413 11.5.2.4 Action ADV_IMP.2.1E

7414 ISO/IEC 15408-3 ADV_IMP.2.1C *The implementation representation shall define the TSF to*
7415 *a level of detail such that the TSF can be generated without further design decisions.*

7416 11.5.2.4.1 Work unit ADV_IMP.2-1

7417 The evaluator **shall check** that the implementation representation defines the TSF to a level of
7418 detail such that the TSF can be generated without further design decisions.

7419 Source code or hardware diagrams and/or IC hardware design language code or layout data that
7420 are used to build the actual hardware are examples of parts of an implementation representation.
7421 The evaluator samples the implementation representation to gain confidence that it is at the
7422 appropriate level and not, for instance, a pseudo-code level which requires additional design
7423 decisions to be made. The evaluator is encouraged to perform a quick check when first looking at
7424 the implementation representation to assure themselves that the developer is on the right track.
7425 However, the evaluator is also encourage to perform the bulk of this check while working on other
7426 work units that call for examining the implementation; this will ensure the sample examined for
7427 this work unit is relevant.

7428 If the evaluator has the possibility to actually execute or witness the "built" procedure used to
7429 transfer the implementation representation into the actual implementation, and to compare the
7430 result to the TOE as delivered, this may provide an easier and at the same time more reliable check
7431 for this work unit (and possibly also for the following one).

7432 ISO/IEC 15408-3 ADV_IMP.2.2C *The implementation representation shall be in the form*
7433 *used by the development personnel.*

7434 11.5.2.4.2 Work unit ADV_IMP.2-2

7435 The evaluator **shall check** that the implementation representation is in the form used by
7436 development personnel.

7437 The implementation representation is manipulated by the developer in form that it suitable for
7438 transformation to the actual implementation. For instance, the developer may work with files
7439 containing source code, which is eventually compiled to become part of the TSF. The developer
7440 makes available the implementation representation in the form they use, so that the evaluator may
7441 use automated techniques in the analysis. This also increases the confidence that the
7442 implementation representation examined is actually the one used in the production of the TSF (as
7443 opposed to the case where it is supplied in an alternate presentation format, such as a word
7444 processor document). It should be noted that other forms of the implementation representation
7445 may also be used by the developer; these forms are supplied as well. The overall goal is to supply
7446 the evaluator with the information that will maximise the evaluator's analysis efforts.

- 7447 The evaluator samples the implementation representation to gain confidence that it is the version
 7448 that is usable by the developer. The sample is such that the evaluator has assurance that all areas of
 7449 the implementation representation are in conformance with the requirement; however, a complete
 7450 examination of the entire implementation representation is unnecessary.
- 7451 Conventions in some forms of the implementation representation may make it difficult or
 7452 impossible to determine from just the implementation representation itself what the actual result
 7453 of the compilation or run-time interpretation will be. For example, compiler directives for C
 7454 language compilers will cause the compiler to exclude or include entire portions of the code.
- 7455 Some forms of the implementation representation may require additional information because
 7456 they introduce significant barriers to understanding and analysis. Examples include shrouded
 7457 source code or source code that has been obfuscated in other ways such that it prevents
 7458 understanding and/or analysis. These forms of implementation representation typically result
 7459 from by taking a version of the implementation representation that is used by the TOE developer
 7460 and running a shrouding or obfuscation program on it. While the shrouded representation is what
 7461 is compiled and may be closer to the implementation (in terms of structure) than the original, un-
 7462 shrouded representation, supplying such obfuscated code may cause significantly more time to be
 7463 spent in analysis tasks involving the representation. When such forms of representation are
 7464 created, the components require details on the shrouding tools/algorithms used so that the un-
 7465 shrouded representation can be supplied, and the additional information can be used to gain
 7466 confidence that the shrouding process does not compromise any security mechanisms.
- 7467 The evaluator samples the implementation representation to gain confidence that all of the
 7468 information needed to interpret the implementation representation has been supplied. Note that
 7469 the tools are among those referenced by Tools and techniques (ALC_TAT) components. The
 7470 evaluator is encouraged to perform a quick check when first looking at the implementation
 7471 representation to assure themselves that the developer is on the right track. However, the
 7472 evaluator is also encouraged to perform the bulk of this check while working on other work units
 7473 that call for examining the implementation; this will ensure the sample examined for this work unit
 7474 is relevant.
- 7475 ISO/IEC 15408-3 ADV_IMP.2.3C *The mapping between the TOE design description and the*
 7476 *entire implementation representation shall demonstrate their correspondence.*
- 7477 **11.5.2.4.3 Work unit ADV_IMP.2-3**
- 7478 The evaluator ***shall examine*** the mapping between the TOE design description and the entire
 7479 implementation representation to determine that it is accurate.
- 7480 The evaluator augments the determination of existence (specified in work unit ADV_IMP.2-1) by
 7481 verifying the accuracy of the implementation representation and the TOE design description. For
 7482 those parts of TOE design description that are interesting, the evaluator would verify the
 7483 implementation representation accurately reflects the description provided in the TOE design
 7484 description.
- 7485 For example, the TOE design description might identify a login module that is used to identify and
 7486 authenticate users. If user authentication is sufficiently significant, the evaluator would verify that
 7487 the corresponding code in fact implements that service as described in the TOE design description.
 7488 It might also be worthwhile to verify that the code accepts the parameters as described in the
 7489 functional specification.
- 7490 Usually it will be expected that the evaluator considers at least the functionality required by the
 7491 SFRs chosen in the ST and aspects described in the security architecture description as
 7492 "interesting" in the sense discussed above. Note however that not all aspects of the security
 7493 architecture are necessarily traceable to specific parts of the implementation representation.

7494 It is worth pointing out the developer must perform the mapping for the entire implementation
7495 representation, thereby guaranteeing that the chosen sample will be covered.

7496 **11.5.2.4.4 Work unit ADV_IMP.2-4**

7497 The evaluator *shall examine* the mapping between the TOE design description and the entire
7498 implementation representation to determine that it is complete.

7499 Note that the completeness here is relevant in both directions: The complete TOE design needs to
7500 be covered by the implementation representation and all parts of the implementation
7501 representation needs to be mapped to a corresponding part of the TOE design.

7502 In order to confirm that the entire implementation representation is covered by the mapping the
7503 evaluator will not need to examine the content of every part of the implementation representation.
7504 If (in the case of a software TOE) the mapping is for example described by mapping each source
7505 code file to a module in the TOE design description, it will be sufficient if this mapping is plausible
7506 from the role of the source code file the evaluator can conclude from information like the naming of
7507 the source code files, their grouping in subdirectories or their grouping in "built" procedures. Note,
7508 that aspects of accuracy are covered by the preceding work unit.

7509 In order to confirm that the entire design description is covered by the implementation, the
7510 evaluator may either use a similar argument as in the other direction, i. e. that all modules
7511 contained in the TOE design description are mapped to parts of the implementation representation
7512 in a plausible way. In addition, if the evaluator has established in the preceding work unit that all
7513 SFRs and all applicable parts of the security architecture description are traceable to the
7514 implementation representation this may be seen as sufficient evidence that the mapping is
7515 complete.

7516 **11.6 TSF internals (ADV_INT)**

7517 **11.6.1 Evaluation of sub-activity (ADV_INT.1)**

7518 **11.6.1.1 Objectives**

7519 The objective of this sub-activity is to determine whether the defined subset of the TSF is designed
7520 and structured such that the likelihood of flaws is reduced and that maintenance can be more
7521 readily performed without the introduction of flaws.

7522 **11.6.1.2 Input**

7523 The evaluation evidence for this sub-activity is:

- 7524 a) the ST;
- 7525 b) the TOE design description;
- 7526 c) the implementation representation (if ADV_IMP is part of the claimed assurance);
- 7527 d) the TSF internals description and justification;
- 7528 e) the documentation of the coding standards, as resulting from ALC_TAT.

7529 **11.6.1.3 Application notes**

7530 The role of the internals description is to provide evidence of the structure of the design and
7531 implementation of the TSF.

7532 The structure of the design has two aspects: the constituent parts of the TSF and the procedures
 7533 used to design the TSF. In cases where the TSF is designed in a manner consistent with the design
 7534 represented by the TOE design (see ADV_TDS), the assessment of the TSF design is obvious. In
 7535 cases where the design procedures (see ALC_TAT) are being followed, the assessment of the TSF
 7536 design procedures is similarly obvious.

7537 **11.6.1.4 In cases where the TSF is implemented using procedure-based software, this**
 7538 **structure is assessed on the basis of its *modularity*; the modules identified in the internals**
 7539 **description are the same as the modules identified in the TOE design (Objectives**

7540 **11.6.1.4** The objectives of this sub-activity are to determine whether the formal security policy model of
 7541 the TSF clearly and consistently describes the rules and characteristics of the security policies
 7542 and whether this description corresponds with the description of security functions in the
 7543 functional specification.

7544 **11.6.1.4 Input**

7545 **11.6.1.4** The evaluation evidence for this sub-activity is:

7546 **11.6.1.4** the ST;

7547 **11.6.1.4** the functional specification;

7548 **11.6.1.4** formal security policy model (ADV_SPM.1.1D);

7549 **11.6.1.4** formal proof of correspondence between the model and any formal functional specification
 7550 (ADV_SPM.1.3D);

7551 **11.6.1.4** demonstration of correspondence between the model and the functional specification
 7552 (ADV_SPM.1.4D).

7553 **11.6.1.4 Application notes**

7554 **11.6.1.4** This activity applies to cases where the developer has provided a formal security policy model of
 7555 the TOE.

7556 **11.6.1.4** A formal TOE security policy model is a representation of the rules (synonymously termed
 7557 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
 7558 Their formal counterparts are called security properties and security features, respectively. The
 7559 representation includes but is not limited to algebraic specifications, finite state machines and
 7560 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
 7561 model is accompanied by an informal interpretation explaining how the rules and characteristics
 7562 are mapped to the respective properties and features.

7563 **11.6.1.4** The creation of a formal security policy model helps to identify and eliminate ambiguous,
 7564 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
 7565 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
 7566 of how well the developer has understood the security functionality being implemented and
 7567 whether there are inconsistencies between the security requirements and the TOE design. The
 7568 confidence in the model is accompanied by a proof that it contains no inconsistencies.

7569 **11.6.1.4** A formal security model is a precise formal presentation of the important aspects of security and
 7570 their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that
 7571 defines the TOE security policy and the set of practises (characteristics) that regulates how the
 7572 TSF manages, protects, and otherwise controls the system resources. The model includes the set
 7573 of restrictions and properties that specify how information and computing resources are
 7574 prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
 7575 arguments showing that these restrictions and properties play a key role in the enforcement of

- 7576 the SFRs. It consists both of the formalisms that express the security functionality, as well as
 7577 ancillary text to explain the model and to provide it with context. The security behaviour of the
 7578 TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
 7579 the TOE and with its operational environment), as well as its internal behaviour.
- 7580 **11.6.1.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
 7581 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 7582 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 7583 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 7584 are always prone to fallacies; especially if relationships among subjects, objects and operations
 7585 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 7586 characteristics of the security policy model are mapped to respective properties and features
 7587 within some formal system, whose rigour and strength can afterwards be used to obtain the
 7588 security properties by means of theorems and formal proof.
- 7589 **11.6.1.4** While the term "formal security policy model" is used in academic circles, the CC's
 7590 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
 7591 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 7592 of SFRs being claimed.
- 7593 **11.6.1.4** The term security policy has traditionally been associated with only access control
 7594 policies, whether label-based (mandatory access control) or user-based (discretionary access
 7595 control). However, a security policy is not limited to access control; there are also audit policies,
 7596 identification policies, authentication policies, encryption policies, management policies, and any
 7597 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 7598 contains an assignment for identifying these policies that are formally modelled.
- 7599 **11.6.1.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
 7600 because either a given policy can not be formally modelled in the otherwise well suited
 7601 framework, or because the nature of the TOE renders impossible the modelling of policies that
 7602 would otherwise be possible to model.
- 7603 **11.6.1.4 Action ADV_SPM.1.1E**
- 7604 **11.6.1.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 7605 *text as required, and identify the security policies of the TSF that are modelled.*
- 7606 **11.6.1.4 Work unit ADV_SPM.1-1**
- 7607 **11.6.1.4** The evaluator ***shall examine the TOE security policy model to determine that it is***
 7608 *written in a formal style.*
- 7609 **11.6.1.4** The evaluator identifies the formal framework upon which the TOE security policy
 7610 model is based and ensures that it is founded on well established mathematical concepts. **They**
 7611 **also identify the security properties and features addressed in the application notes and ensure**
 7612 **the formalization of at least one security policy.**
- 7613 **11.6.1.4** For guidance on formal methods refer to ISO/IEC 15408-3
- 7614 **11.6.1.4 Work unit ADV_SPM.1-2**
- 7615 **11.6.1.4** The evaluator ***shall examine the TOE security policy model to determine that it***
 7616 *contains all necessary informal explanatory text.*
- 7617 **11.6.1.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
 7618 to make clear the meaning of any formal notation and how they are used) including the security
 7619 properties and features.

7620	11.6.1.4 Work unit ADV_SPM.1-3
7621	11.6.1.4 The evaluator <i>shall examine the TOE security policy model to determine that all</i> security policies of the TSF are identified that are modelled.
7622	
7623	
7624	11.6.1.4 The evaluator determines whether the SPM identifies the security policies for which a model is provided, identifying the relevant portions of the statement of SFRs that comprise each of the modelled policies.
7625	
7626	
7627	11.6.1.4 The evaluator determines whether the list of security policies identified by the SPM is consistent with the assignment of ADV_SPM.1.1D in the ST.
7628	
7629	
7630	11.6.1.4 ADV_SPM.1.2C <i>For all policies that are modelled, the model shall define security for the TOE and provide a formal proof that the TOE cannot reach a state that is not secure.</i>
7631	
7632	
7633	11.6.1.4 Work unit ADV_SPM.1-4
7634	11.6.1.4 The evaluator <i>shall examine the principles and characteristics of the security policies</i> to determine that the modelled security behaviour of the TOE is clearly articulated.
7635	
7636	11.6.1.4 The security policies are expressed in terms of security principles (rules) which are modelled by security properties and define the secure state of the TOE. For example, a model based on state transitions could describe the security policies in terms of principles of its states, identify its initial state, and define what it means to be a secure state.
7637	
7638	
7639	
7640	11.6.1.4 The evaluator determines that the security policies are reflected within their formal counterparts of the TSP model.
7641	
7642	11.6.1.4 The TOE security behaviour is expressed in terms of security characteristics (i.e. portions of TOE security functionality managing, protecting, and otherwise controlling the system resources including attributes and conditions of the TOE) which are modelled by security features. For example, a model based on state transitions could describe the characteristics as possible actions in each secure state in a level of detail sufficient to decide into which state the TOE will be transformed by that action.
7643	
7644	
7645	
7646	
7647	
7648	11.6.1.4 Together the security principles and characteristics describe the entire security posture of the TOE.
7649	
7650	11.6.1.4 In the context of a formal TOE security policy model the security behaviour is considered to be clearly articulated only if an adequate mapping from principles and characteristics to their respective formal counterparts properties and features has been given. The mapping is considered to be adequate if the level of abstraction from the TOE's realization is detailed enough to allow for correct identification of all security objectives and the relation to the security environment.
7651	
7652	
7653	
7654	
7655	
7656	11.6.1.4 The above condition for clear articulation is necessary but not sufficient. An informal interpretation of all formal concepts (including attributes, predicates and variables, if available) must be provided in order to make clear their intended meaning.
7657	
7658	
7659	11.6.1.4 Work unit ADV_SPM.1-5
7660	11.6.1.4 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i> it formally proves that the security features enforce the security properties.
7661	

- 7662 **11.6.1.4** To determine the enforcement, the evaluator considers the security properties and the
 7663 security features and verifies that the arguments used in the proof are valid. The proof of
 7664 correspondence between the security properties and the security features shall be formal.
- 7665 **11.6.1.4** The validity of the security properties shall mean that the TOE is in a secure state. By
 7666 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 7667 state.
- 7668 **11.6.1.4 Work unit ADV_SPM.1-6**
- 7669 **11.6.1.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
 7670 *it proves the internal consistency of the TOE security policy model.*
- 7671 **11.6.1.4** The proof shall show the absence of contradictions within the TOE security policy
 7672 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 7673 used in the proof are valid.
- 7674 **11.6.1.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
 7675 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 7676 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 7677 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 7678 security policy model that prove the internal consistency by means of a combination with generic
 7679 arguments of the formal framework.
- 7680 **11.6.1.4 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 7681 *specification shall be at the correct level of formality.*
- 7682 **11.6.1.4 Work unit ADV_SPM.1-7**
- 7683 **11.6.1.4** The evaluator *shall examine the correspondence between the model and the functional*
 7684 *specification to determine that a semiformal demonstration of correspondence between the*
 7685 *model and any semiformal functional specification is provided.*
- 7686 **11.6.1.4** This work unit is only applicable to a semiformal presentation of the functional
 7687 specification, which is required by ADV_FSP.5.2C.
- 7688 **11.6.1.4** A semiformal correspondence is one that results from a structured approach with a
 7689 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 7690 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 7691 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 7692 **11.6.1.4** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 7693 **methods**'.
- 7694 **11.6.1.4 Work unit ADV_SPM.1-8**
- 7695 **11.6.1.4** The evaluator *shall examine the correspondence between the model and the functional*
 7696 *specification to determine that a formal proof of correspondence between the model and any*
 7697 *formal functional specification is provided.*
- 7698 **11.6.1.4** This work unit is only applicable to a formal presentation of the functional specification,
 7699 which is required by ADV_FSP.6.2D.
- 7700 **11.6.1.4** There should be a formal proof of correspondence between the model and any formal
 7701 functional specification.
- 7702 **11.6.1.4** The formal proof of correspondence removes all subjective interpretations of its terms
 7703 by enlisting well-established mathematical concepts to define the syntax and semantics of the

- 7704 formal notation and uses rules that support logical reasoning. The security features within the
7705 TOE (which are identified in the formal TSP model) are expressed in a formal specification
7706 language and shown to be satisfied by the formal specification.
- 7707 **11.6.1.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 7708 **11.6.1.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
7709 *specification is consistent and complete with respect to the model.*
- 7710 **11.6.1.4 Work unit ADV_SPM.1-9**
- 7711 **11.6.1.4** The evaluator **shall examine the correspondence to determine that the behaviour at the**
7712 **TSF interfaces (as articulated in the functional specification) is complete with respect to the**
7713 **behaviour modelled by the security features.**
- 7714 **11.6.1.4** The term “correspondence” here means both the formal proof of correspondence
7715 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
7716 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 7717 **11.6.1.4** In determining completeness of the correspondence, the evaluator considers the
7718 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
7719 features of the TSP model. The demonstration should show that all characteristics belonging to
7720 policies that are required to be modelled have an associated feature description in the TOE
7721 security policy model, and that each feature of the TSP model does occur in the mapping.
- 7722 **11.6.1.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
7723 developer’s side (also confer the application notes above).
- 7724 **11.6.1.4 Work unit ADV_SPM.1-10**
- 7725 **11.6.1.4** The evaluator **shall examine the correspondence to determine that the behaviour at the**
7726 **TSF interfaces (as articulated in the functional specification) is consistent with respect to the**
7727 **behaviour modelled by the security features.**
- 7728 **11.6.1.4** The term “correspondence” here means both the formal proof of correspondence
7729 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
7730 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 7731 **11.6.1.4** The meaning of consistency reflects the conventional understanding in contrast to the
7732 internal consistency concept of work unit ADV_SPM.1-6.
- 7733 **11.6.1.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
7734 security features established in the preceding work unit and verifies that the correspondence
7735 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
7736 behaviour.
- 7737 **11.6.1.4** For example, if TSFI behaviour dealt with access management on the granularity of
7738 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
7739 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
7740 management for groups of users, then a TSP model describing the security behaviour of the TOE
7741 in terms of individual users would also not be consistent.
- 7742 **11.6.1.4** As another example, if remote untrusted users had to pass more stringent
7743 authentication procedures than administrators whose only point of access were within a
7744 physically-protected area, then this difference in authentication procedures had to be reflected in
7745 the security features.

7746 **11.6.1.4** TOE design (ADV_TDS)). A module consists of one or more source code files that cannot
7747 be decomposed into smaller compilable units.

7748 The use of the assignment in this component levies stricter constraints on the subset of the TSF
7749 that is explicitly identified in the assignment ADV_INT.1.1D than on the remainder of the TSF.
7750 While the entire TSF is to be designed using good engineering principles and result in a well-
7751 structured TSF, only the specified subset is specifically analysed for this characteristic. The
7752 evaluator determines that the developer's application of coding standards result in a TSF that is
7753 understandable.

7754 The primary goal of this component is to ensure the TSF subset's implementation representation is
7755 understandable to facilitate maintenance and analysis (of both the developer and evaluator).

7756 **11.6.1.8 Action ADV_INT.1.1E**

7757 ISO/IEC 15408-3 ADV_INT.1.1C: *The justification shall explain the characteristics used to judge the*
7758 *meaning of "well-structured".*

7759 **11.6.1.8.1 Work unit ADV_INT.1-1**

7760 The evaluator **shall examine** the justification to determine that it identifies the basis for
7761 determining whether the TSF is well-structured.

7762 The evaluator verifies that the criteria for determining the characteristic of being well-structured
7763 are clearly defined in the justification. Acceptable criteria typically originate from industry
7764 standards for the technology discipline. For example, procedural software that executes linearly is
7765 traditionally viewed as well-structured if it adheres to software engineering programming
7766 practises, such as those defined in the IEEE Standard (*IEEE Std 610.12-1990*). For example, it would
7767 identify the criteria for the procedural software portions of the TSF subset:

7768 a) the process used for modular decomposition

7769 b) coding standards used in the development of the implementation

7770 c) a description of the maximum acceptable level of intermodule coupling exhibited by the
7771 TSF subset

7772 d) a description of the minimum acceptable level of cohesion exhibited the modules of the
7773 TSF subset

7774 For other types of technologies used in the TOE - such as non-procedural software (e.g. object-
7775 oriented programming), widespread commodity hardware (e.g. PC microprocessors), and special-
7776 purpose hardware (e.g. smart-card processors) - the evaluator should seek guidance from the
7777 evaluation authority for determining the adequacy of criteria for being "well-structured".

7778 ISO/IEC 15408-3 ADV_INT.1.2C: *The TSF internals description shall demonstrate that the assigned*
7779 *subset of the TSF is well-structured.*

7780 **11.6.1.8.2 Work unit ADV_INT.1-2**

7781 The evaluator **shall check** the TSF internals description to determine that it identifies the Assigned
7782 subset of the TSF.

7783 This subset may be identified in terms of the internals of the TSF at any layer of abstraction. For
7784 example, it may be in terms of the structural elements of the TSF as identified in the TOE design
7785 (e.g. the audit subsystem), or in terms of the implementation (e.g. *encrypt.c* and *decrypt.c* files, or
7786 the 6227 IC chip).

7787 It is insufficient to identify this subset in terms of the claimed SFRs (e.g. the portion of the TSF that
7788 provide anonymity as defined in FPR_ANO.2) because this does not indicate where to focus the
7789 analysis.

7790 **11.6.1.8.3 Work unit ADV_INT.1-3**

7791 The evaluator ***shall examine*** the TSF internals description to determine that it demonstrates that
7792 the assigned TSF subset is well-structured.

7793 The evaluator examines the internals description to ensure that it provides a sound explanation of
7794 how the TSF subset meets the criteria from ADV_INT.1-1

7795 For example, it would explain how the procedural software portions of the TSF subset meets the
7796 following:

7797 a) that there is a one-to-one correspondence between the modules identified in the TSF
7798 subset and the modules described in the TOE design (ADV_TDS)

7799 b) how the TSF design is a reflection of the modular decomposition process

7800 c) a justification for all instances where the coding standards were not used or met

7801 d) a justification for any coupling or cohesion outside the acceptable bounds

7802 **11.6.1.9 Action ADV_INT.1.2E**

7803 **11.6.1.9.1 Work unit ADV_INT.1-4**

7804 The evaluator ***shall determine*** that the TOE design for the assigned TSF subset is well-structured.

7805 The evaluator examines a sample of the TOE design to verify the accuracy of the justification. For
7806 example, a sample of the TOE design is analysed to determine its adherence to the design
7807 standards, etc. As with all areas where the evaluator performs activities on a subset the evaluator
7808 provides a justification of the sample size and scope

7809 The description of the TOE's decomposition into subsystems and modules will make the argument
7810 that the TSF subset is well-structured self-evident. Verification that the procedures for structuring
7811 the TSF (as examined in ALC_TAT) are being followed will make it self-evident that the TSF subset
7812 is well-structured.

7813 **11.6.1.9.2 Work unit ADV_INT.1-5**

7814 The evaluator ***shall determine*** that the assigned TSF subset is well-structured.

7815 If ADV_IMP is not part of the claimed assurance, then this work unit is not applicable and is
7816 therefore considered to be satisfied.

7817 The evaluator examines a sample of the TSF subset to verify the accuracy of the internals
7818 description. For example, a sample of the procedural software portions of the TSF subset is
7819 analysed to determine its cohesion and coupling, its adherence to the coding standards, etc. As with
7820 all areas where the evaluator performs activities on a subset the evaluator provides a justification
7821 of the sample size and scope.

7822 **11.6.2 Evaluation of sub-activity (ADV_INT.2)**

7823 **11.6.2.1 Objectives**

7824 The objective of this sub-activity is to determine whether the TSF is designed and structured such
7825 that the likelihood of flaws is reduced and that maintenance can be more readily performed
7826 without the introduction of flaws.

7827 **11.6.2.2 Input**

7828 The evaluation evidence for this sub-activity is:

- 7829 a) the modular design description;
- 7830 b) the implementation representation (if ADV_IMP is part of the claimed assurance));
- 7831 c) the TSF internals description;
- 7832 d) the documentation of the coding standards, as resulting from ALC_TAT.

7833 **11.6.2.3 Application notes**

7834 The role of the internals description is to provide evidence of the structure of the design and
7835 implementation of the TSF.

7836 The structure of the design has two aspects: the constituent parts of the TSF and the procedures
7837 used to design the TSF. In cases where the TSF is designed in a manner consistent with the design
7838 represented by the TOE design (see ADV_TDS), the assessment of the TSF design is obvious. In
7839 cases where the design procedures (see ALC_TAT) are being followed, the assessment of the TSF
7840 design procedures is similarly obvious.

7841 **11.6.2.4 In cases where the TSF is implemented using procedure-based software, this structure is assessed**
7842 **on the basis of its *modularity*; the modules identified in the internals description are the same as**
7843 **the modules identified in the TOE design (Objectives)**

7844 **11.6.2.4** The objectives of this sub-activity are to determine whether the formal security policy model of
7845 the TSF clearly and consistently describes the rules and characteristics of the security policies
7846 and whether this description corresponds with the description of security functions in the
7847 functional specification.

7848 **11.6.2.4 Input**

7849 **11.6.2.4** The evaluation evidence for this sub-activity is:

- 7850 **11.6.2.4** the ST;
- 7851 **11.6.2.4** the functional specification;
- 7852 **11.6.2.4** formal security policy model (ADV_SPM.1.1D);
- 7853 **11.6.2.4** formal proof of correspondence between the model and any formal functional specification
7854 (ADV_SPM.1.3D);
- 7855 **11.6.2.4** demonstration of correspondence between the model and the functional specification
7856 (ADV_SPM.1.4D).

7857 **11.6.2.4 Application notes**

7858 **11.6.2.4** This activity applies to cases where the developer has provided a formal security policy
7859 model of the TOE.

7860 **11.6.2.4** A formal TOE security policy model is a representation of the rules (synonymously
7861 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
7862 terms. Their formal counterparts are called security properties and security features,
7863 respectively. The representation includes but is not limited to algebraic specifications, finite state
7864 machines and logic formalisms strong enough to formally infer the properties from the features.
7865 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
7866 characteristics are mapped to the respective properties and features.

7867 **11.6.2.4** The creation of a formal security policy model helps to identify and eliminate
7868 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
7869 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
7870 judgement of how well the developer has understood the security functionality being
7871 implemented and whether there are inconsistencies between the security requirements and the
7872 TOE design. The confidence in the model is accompanied by a proof that it contains no
7873 inconsistencies.

7874 **11.6.2.4** A formal security model is a precise formal presentation of the important aspects of
7875 security and their relationship to the behaviour of the TOE; it identifies the set of rules
7876 (principles) that defines the TOE security policy and the set of practises (characteristics) that
7877 regulates how the TSF manages, protects, and otherwise controls the system resources. The
7878 model includes the set of restrictions and properties that specify how information and computing
7879 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
7880 engineering arguments showing that these restrictions and properties play a key role in the
7881 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
7882 as well as ancillary text to explain the model and to provide it with context. The security
7883 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
7884 with the rest of the TOE and with its operational environment), as well as its internal behaviour.

7885 **11.6.2.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
7886 considering the proposed security requirements of the ST. The informal abstraction is taken to be
7887 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
7888 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
7889 are always prone to fallacies; especially if relationships among subjects, objects and operations
7890 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
7891 characteristics of the security policy model are mapped to respective properties and features
7892 within some formal system, whose rigour and strength can afterwards be used to obtain the
7893 security properties by means of theorems and formal proof.

7894 **11.6.2.4** While the term “formal security policy model” is used in academic circles, the CC's
7895 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
7896 claimed. Therefore, the formal security policy model is merely a formal representation of the set
7897 of SFRs being claimed.

7898 **11.6.2.4** The term security policy has traditionally been associated with only access control
7899 policies, whether label-based (mandatory access control) or user-based (discretionary access
7900 control). However, a security policy is not limited to access control; there are also audit policies,
7901 identification policies, authentication policies, encryption policies, management policies, and any
7902 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
7903 contains an assignment for identifying these policies that are formally modelled.

7904 **11.6.2.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
7905 because either a given policy can not be formally modelled in the otherwise well suited

- 7906 framework, or because the nature of the TOE renders impossible the modelling of policies that
7907 would otherwise be possible to model.
- 7908 **11.6.2.4 Action ADV_SPM.1.1E**
- 7909 **11.6.2.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
7910 *text as required, and identify the security policies of the TSF that are modelled.*
- 7911 **11.6.2.4 Work unit ADV_SPM.1-1**
- 7912 **11.6.2.4** The evaluator *shall examine the TOE security policy model to determine that it is*
7913 *written in a formal style.*
- 7914 **11.6.2.4** The evaluator identifies the formal framework upon which the TOE security policy
7915 model is based and ensures that it is founded on well established mathematical concepts. **They**
7916 **also identify the security properties and features addressed in the application notes and ensure**
7917 **the formalization of at least one security policy.**
- 7918 **11.6.2.4** For guidance on formal methods refer to ISO/IEC 15408-3
- 7919 **11.6.2.4 Work unit ADV_SPM.1-2**
- 7920 **11.6.2.4** The evaluator *shall examine the TOE security policy model to determine that it*
7921 *contains all necessary informal explanatory text.*
- 7922 **11.6.2.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
7923 to make clear the meaning of any formal notation and how they are used) including the security
7924 properties and features.
- 7925 **11.6.2.4 Work unit ADV_SPM.1-3**
- 7926 **11.6.2.4** The evaluator *shall examine the TOE security policy model to determine that all*
7927 *security policies of the TSF are identified that are modelled.*
- 7928 **11.6.2.4** The evaluator determines whether the SPM identifies the security policies for which a
7929 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
7930 of the modelled policies.
- 7931 **11.6.2.4** The evaluator determines whether the list of security policies identified by the SPM is
7932 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 7933 **11.6.2.4** The evaluator determines whether for each security policy identified by the SPM a
7934 model is in fact provided.
- 7935 **11.6.2.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
7936 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
7937 *not secure.*
- 7938 **11.6.2.4 Work unit ADV_SPM.1-4**
- 7939 **11.6.2.4** The evaluator *shall examine the principles and characteristics of the security policies*
7940 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 7941 **11.6.2.4** The security policies are expressed in terms of security principles (rules) which are
7942 modelled by security properties and define the secure state of the TOE. For example, a model
7943 based on state transitions could describe the security policies in terms of principles of its states,
7944 identify its initial state, and define what it means to be a secure state.

- 7945 **11.6.2.4** The evaluator determines that the security policies are reflected within their formal
7946 counterparts of the TSP model.
- 7947 **11.6.2.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
7948 portions of TOE security functionality managing, protecting, and otherwise controlling the system
7949 resources including attributes and conditions of the TOE) which are modelled by security
7950 features. For example, a model based on state transitions could describe the characteristics as
7951 possible actions in each secure state in a level of detail sufficient to decide into which state the
7952 TOE will be transformed by that action.
- 7953 **11.6.2.4** Together the security principles and characteristics describe the entire security posture
7954 of the TOE.
- 7955 **11.6.2.4** In the context of a formal TOE security policy model the security behaviour is
7956 considered to be clearly articulated only if an adequate mapping from principles and
7957 characteristics to their respective formal counterparts properties and features has been given.
7958 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
7959 detailed enough to allow for correct identification of all security objectives and the relation to the
7960 security environment.
- 7961 **11.6.2.4** The above condition for clear articulation is necessary but not sufficient. An informal
7962 interpretation of all formal concepts (including attributes, predicates and variables, if available)
7963 must be provided in order to make clear their intended meaning.
- 7964 **11.6.2.4 Work unit ADV_SPM.1-5**
- 7965 **11.6.2.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
7966 *it formally proves that the security features enforce the security properties.*
- 7967 **11.6.2.4** To determine the enforcement, the evaluator considers the security properties and the
7968 security features and verifies that the arguments used in the proof are valid. The proof of
7969 correspondence between the security properties and the security features shall be formal.
- 7970 **11.6.2.4** The validity of the security properties shall mean that the TOE is in a secure state. By
7971 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
7972 state.
- 7973 **11.6.2.4 Work unit ADV_SPM.1-6**
- 7974 **11.6.2.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
7975 *it proves the internal consistency of the TOE security policy model.*
- 7976 **11.6.2.4** The proof shall show the absence of contradictions within the TOE security policy
7977 model. In determining the absence of contradictions, the evaluator verifies that the arguments
7978 used in the proof are valid.
- 7979 **11.6.2.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
7980 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
7981 security policy model usually is not possible due to the fundamental nature of formal frameworks.
7982 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
7983 security policy model that prove the internal consistency by means of a combination with generic
7984 arguments of the formal framework.
- 7985 **11.6.2.4 ADV_SPM.1.3C** ***The correspondence between the model and the functional***
7986 ***specification shall be at the correct level of formality.***

- 7987 **11.6.2.4 Work unit ADV_SPM.1-7**
- 7988 **11.6.2.4** The evaluator *shall examine the correspondence between the model and the*
 7989 functional specification to determine that a semiformal demonstration of correspondence
 7990 between the model and any semiformal functional specification is provided.
- 7991 **11.6.2.4** This work unit is only applicable to a semiformal presentation of the functional
 7992 specification, which is required by ADV_FSP.5.2C.
- 7993 **11.6.2.4** A semiformal correspondence is one that results from a structured approach with a
 7994 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 7995 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 7996 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 7997 **11.6.2.4** For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
 7998 methods’.
- 7999 **11.6.2.4 Work unit ADV_SPM.1-8**
- 8000 **11.6.2.4** The evaluator *shall examine the correspondence between the model and the functional*
 8001 specification to determine that a formal proof of correspondence between the model and any
 8002 formal functional specification is provided.
- 8003 **11.6.2.4** This work unit is only applicable to a formal presentation of the functional specification,
 8004 which is required by ADV_FSP.6.2D.
- 8005 **11.6.2.4** There should be a formal proof of correspondence between the model and any formal
 8006 functional specification.
- 8007 **11.6.2.4** The formal proof of correspondence removes all subjective interpretations of its terms
 8008 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 8009 formal notation and uses rules that support logical reasoning. The security features within the
 8010 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 8011 language and shown to be satisfied by the formal specification.
- 8012 **11.6.2.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 8013 **11.6.2.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 8014 *specification is consistent and complete with respect to the model.*
- 8015 **11.6.2.4 Work unit ADV_SPM.1-9**
- 8016 **11.6.2.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 8017 TSF interfaces (as articulated in the functional specification) is complete with respect to the
 8018 behaviour modelled by the security features.
- 8019 **11.6.2.4** The term “correspondence” here means both the formal proof of correspondence
 8020 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 8021 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 8022 **11.6.2.4** In determining completeness of the correspondence, the evaluator considers the
 8023 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 8024 features of the TSP model. The demonstration should show that all characteristics belonging to
 8025 policies that are required to be modelled have an associated feature description in the TOE
 8026 security policy model, and that each feature of the TSP model does occur in the mapping.
- 8027 **11.6.2.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
 8028 developer’s side (also confer the application notes above).

8029 **11.6.2.4 Work unit ADV_SPM.1-10**

8030 **11.6.2.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 8031 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 8032 behaviour modelled by the security features.

8033 **11.6.2.4** The term “correspondence” here means both the formal proof of correspondence
 8034 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 8035 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

8036 **11.6.2.4** The meaning of consistency reflects the conventional understanding in contrast to the
 8037 internal consistency concept of work unit ADV_SPM.1-6.

8038 **11.6.2.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 8039 security features established in the preceding work unit and verifies that the correspondence
 8040 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 8041 behaviour.

8042 **11.6.2.4** For example, if TSFI behaviour dealt with access management on the granularity of
 8043 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 8044 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 8045 management for groups of users, then a TSP model describing the security behaviour of the TOE
 8046 in terms of individual users would also not be consistent.

8047 **11.6.2.4** As another example, if remote untrusted users had to pass more stringent
 8048 authentication procedures than administrators whose only point of access were within a
 8049 physically-protected area, then this difference in authentication procedures had to be reflected in
 8050 the security features.

8051 **11.6.2.4** TOE design (ADV_TDS)). A module consists of one or more source code files that cannot
 8052 be decomposed into smaller compilable units.

8053 The primary goal of this component is to ensure the TSF's implementation representation is
 8054 understandable to facilitate maintenance and analysis (of both the developer and evaluator).

8055 **11.6.2.8 Action ADV_INT.2.1E**

8056 ISO/IEC 15408-3 ADV_INT.2.1C: *The justification shall describe the characteristics used to judge the*
 8057 *meaning of “well-structured”.*

8058 **11.6.2.8.1 Work unit ADV_INT.2-1**

8059 The evaluator *shall examine* the justification to determine that it identifies the basis for
 8060 determining whether the TSF is well-structured.

8061 The evaluator verifies that the criteria for determining the characteristic of being well-structured
 8062 are clearly defined in the justification. Acceptable criteria typically originate from industry
 8063 standards for the technology discipline. For example, procedural software that executes linearly is
 8064 traditionally viewed as well-structured if it adheres to software engineering programming
 8065 practises, such as those defined in the IEEE Standard (*IEEE Std 610.12-1990*). For example, it would
 8066 identify the criteria for the procedural software portions of the TSF:

- 8067 a) the process used for modular decomposition
- 8068 b) coding standards used in the development of the implementation
- 8069 c) a description of the maximum acceptable level of intermodule coupling exhibited by the
 8070 TSF

- 8071 d) a description of the minimum acceptable level of cohesion exhibited the modules of the
8072 TSF
- 8073 For other types of technologies used in the TOE - such as non-procedural software (e.g. object-
8074 oriented programming), widespread commodity hardware (e.g. PC microprocessors), and special-
8075 purpose hardware (e.g. smart-card processors) - the evaluation authority should be consulted for
8076 determining the adequacy of criteria for being “well-structured”.
- 8077 ISO/IEC 15408-3 ADV_INT.2.2C: *The TSF internals description shall demonstrate that the entire TSF*
8078 *is well-structured.*
- 8079 **11.6.2.8.2 Work unit ADV_INT.2-2**
- 8080 The evaluator ***shall examine*** the TSF internals description to determine that it demonstrates that
8081 the TSF is well-structured.
- 8082 The evaluator examines the internals description to ensure that it provides a sound explanation of
8083 how the TSF meets the criteria from ADV_INT.2-1
- 8084 For example, it would explain how the procedural software portions of the TSF meet the following:
- 8085 a) that there is a one-to-one correspondence between the modules identified in the TSF and
8086 the modules described in the TOE design (ADV_TDS)
- 8087 b) how the TSF design is a reflection of the modular decomposition process
- 8088 c) a justification for all instances where the coding standards were not used or met
- 8089 d) a justification for any coupling or cohesion outside the acceptable bounds
- 8090 **11.6.2.9 Action ADV_INT.2.2E**
- 8091 **11.6.2.9.1 Work unit ADV_INT.2-3**
- 8092 The evaluator ***shall determine*** that the TOE design is well-structured.
- 8093 The evaluator examines the TOE design of a sample of the TSF to verify the accuracy of the
8094 justification. For example, a sample of the TOE design is analysed to determine its adherence to the
8095 design standards, etc. As with all areas where the evaluator performs activities on a subset the
8096 evaluator provides a justification of the sample size and scope
- 8097 The description of the TOE's decomposition into subsystems and modules will make the argument
8098 that the TSF subset is well-structured self-evident. Verification that the procedures for structuring
8099 the TSF (as examined in ALC_TAT) are being followed will make it self-evident that the TSF subset
8100 is well-structured.
- 8101 **11.6.2.9.2 Work unit ADV_INT.2-4**
- 8102 The evaluator ***shall determine*** that the TSF is well-structured.
- 8103 If ADV_IMP is not part of the claimed assurance, then this work unit is not applicable and is
8104 therefore considered to be satisfied.
- 8105 The evaluator examines a sample of the TSF to verify the accuracy of the internals description. For
8106 example, a sample of the procedural software portions of the TSF is analysed to determine its
8107 cohesion and coupling, its adherence to the coding standards, etc. As with all areas where the
8108 evaluator performs activities on a subset the evaluator provides a justification of the sample size
8109 and scope.

8110 **11.6.3 Evaluation of sub-activity (ADV_INT.3)**

8111 **11.6.3.1 Objectives**

8112 The objective of this sub-activity is to determine whether the TSF is designed and structured such
8113 that the likelihood of flaws is reduced and that maintenance can be more readily performed
8114 without the introduction of flaws.

8115 **11.6.3.2 Input**

8116 The evaluation evidence for this sub-activity is:

- 8117 a) the modular design description;
- 8118 b) the implementation representation (if ADV_IMP is part of the claimed
8119 assurance);
- 8120 c) the TSF internals description;
- 8121 d) the documentation of the coding standards, as resulting from ALC_TAT.

8122 **11.6.3.3 Application notes**

8123 The role of the internals description is to provide evidence of the structure of the design and
8124 implementation of the TSF.

8125 The structure of the design has two aspects: the constituent parts of the TSF and the procedures
8126 used to design the TSF. In cases where the TSF is designed in a manner consistent with the design
8127 represented by the TOE design (see ADV_TDS), the assessment of the TSF design is obvious. In
8128 cases where the design procedures (see ALC_TAT) are being followed, the assessment of the TSF
8129 design procedures is similarly obvious.

8130 In cases where the TSF is implemented using procedure-based software, this structure is assessed
8131 on the basis of its modularity; the modules identified in the internals description are the same as
8132 the modules identified in the TOE design (TOE design (ADV_TDS)). A module consists of one or
8133 more source code files that cannot be decomposed into smaller compilable units.

8134 The primary goal of this component is to ensure the TSF's implementation representation is
8135 understandable to facilitate maintenance and analysis (of both the developer and evaluator).

8136 **11.6.3.4 Action ADV_INT.3.1E**

8137 **ADV_INT.3.1C** *The justification shall describe the characteristics used to judge the*
8138 *meaning of "well-structured" and "complex".*

8139 **11.6.3.4.1 Work unit ADV_INT.3-1**

8140 The evaluator **shall examine** the justification to determine that it identifies the basis for
8141 determining whether the TSF is "well-structured" and "not overly complex".

8142 The evaluator verifies that the criteria for determining the characteristic of being "well-structured"
8143 and "complex" are clearly defined in the justification. Acceptable criteria typically originate from
8144 industry standards for the technology discipline. For example, procedural software that executes
8145 linearly is traditionally viewed as well-structured if it adheres to software engineering
8146 programming practises, such as those defined in the IEEE Standard (IEEE Std 610.12-1990). For
8147 example, it would identify the criteria for the procedural software portions of the TSF:

- 8148 a) the process used for modular decomposition

- 8149 b) coding standards used in the development of the implementation
- 8150 c) a description of the maximum acceptable level of intermodule coupling
- 8151 exhibited by the TSF
- 8152 d) a description of the minimum acceptable level of cohesion exhibited the
- 8153 modules of the TSF
- 8154 Complexity can for example be measured in the number of decision points and logical paths of
- 8155 execution that code takes. Software engineering literature cites complexity as a negative
- 8156 characteristic of software because it impedes understanding of the logic and flow of the code.
- 8157 Another impediment to the understanding of code is the presence of code that is unnecessary, in
- 8158 that it is unused or redundant.
- 8159 1 Design complexity minimisation is a key characteristic of a reference validation
- 8160 mechanism, the purpose of which is to arrive at a TSF that is easily understood so
- 8161 that it can be completely analysed.
- 8162 2 See also CC 3.1, Part 3, Annex A.3 for additional information on TSF internals.
- 8163 3 The consideration in that annex and those made in the preceding paragraphs of this work
- 8164 unit are mainly derived from common knowledge about procedural software. For
- 8165 other types of technologies used in the TOE - such as non-procedural software
- 8166 (e.g. object-oriented programming), widespread commodity hardware (e.g. PC
- 8167 microprocessors), and special-purpose hardware (e.g. smart-card processors) - the
- 8168 evaluation authority should be consulted for determining the adequacy of criteria
- 8169 for being "well-structured" and "not overly complex".
- 8170 4 The evaluator is reminded to be open for plausible definitions given by the developer. If.
- 8171 for example, a smart card developer can justify that the metrics used by him to
- 8172 measure complexity are an industry standard in their field, this should usually be
- 8173 sufficient for acceptance of such metrics.
- 8174 5
- 8175 ISO/IEC 15408-3 ADV_INT.3.2C *The TSF internals description shall demonstrate that the*
- 8176 *entire TSF is well-structured and is not overly complex.*
- 8177 **11.6.3.4.2 Work unit ADV_INT.3-2**
- 8178 The evaluator ***shall examine*** the TSF internals description to determine that it demonstrates that
- 8179 the TSF is well-structured and not overly complex.
- 8180 The evaluator examines the internals description to ensure that it provides a sound explanation of
- 8181 how the TSF meets the criteria from ADV_INT.3-1
- 8182 For example, it would explain how the procedural software portions of the TSF meet the following:
- 8183 a) that there is a one-to-one correspondence between the modules identified in the
- 8184 TSF and the modules described in the TOE design (ADV_TDS)
- 8185 b) how the TSF design is a reflection of the modular decomposition process
- 8186 c) a justification for all instances where the coding standards were not used or met
- 8187 d) a justification for any coupling or cohesion outside the acceptable bounds
- 8188 e) how the modular decomposition process reduces complexity

8189 **11.6.3.5 Action ADV_INT.3.2E**8190 **11.6.3.5.1 Work unit ADV_INT.3-3**

8191 The evaluator ***shall determine*** that the entire TOE design is well-structured and not overly
8192 complex.

8193 The evaluator examines the TOE design description of the TSF to verify the accuracy of the
8194 justification. For example, a sample of the TOE design is analysed to determine its adherence to the
8195 design standards, etc. As with all areas where the evaluator performs activities on a subset the
8196 evaluator provides a justification of the sample size and scope

8197 The description of the TOE's decomposition into subsystems and modules will make the argument
8198 that the TSF is well-structured self-evident. Verification that the procedures for structuring the TSF
8199 (as examined in ALC_TAT) are being followed will make it self-evident that the TSF is well-
8200 structured.

8201 Using the metrics defined by the developer for measuring the complexity of the design will show if
8202 the metrics is met. If the metrics is only defined for the implementation representation and not for
8203 the TOE design (note that adequateness of the metrics was considered already in work unit
8204 ADV_INT.3-1), there may be no need for using the metrics in this work unit, the complexity-issue is
8205 then covered by the next work unit.

8206 **11.6.3.5.2 Work unit ADV_INT.3-4**

8207 The evaluator ***shall determine*** that the entire TSF is well-structured and not overly complex.

8208 If ADV_IMP is not part of the claimed assurance, then this work unit is not applicable and is
8209 therefore considered to be satisfied.

8210 The evaluator examines a sample of the TSF to verify the accuracy of the internals description. For
8211 example, a sample of the procedural software portions of the TSF is analysed to determine its
8212 cohesion and coupling, its adherence to the coding standards, etc. As with all areas where the
8213 evaluator performs activities on a subset the evaluator provides a justification of the sample size
8214 and scope.

8215 Similarly the evaluator applies the metric for complexity as defined by the developer and examined
8216 in work unit ADV_INT.3-1 to either a sample of the implementation representation or the complete
8217 implementation representation (this may depend on the metric) and verifies that the metric is in
8218 fact met. The evaluator may only restrict their application of the metrics to a sample if the
8219 developer has provided the results of the application of the metrics for the entire TSF and the
8220 sampling serves as means to convince the evaluator that the application as done by the developer
8221 was correct (similar to the evaluator's sampling of functional testing already done by the
8222 developer).

8223 **11.7 [PLACE-HOLDER] TOE Modular Traceability of Functional Requirements in**
8224 **Code (ADV_MTC)**

8225 **If the modularity approach included in ASE_AMA, ADV_MTC, ATE_MTK, ATE_MTT remains in**
8226 **ISO/IEC 15408-x then work units will be required to cover these.**

8227 **11.8** Suggestions for text would be welcomed in response to CD1 review. **If none are**
 8228 **received then this topic will be left to the next revision.**Security policy modelling
 8229 **(ADV_SPM)**

8230 **11.8.1 Evaluation of sub-activity (ADV_SPM.1)**

8231 **11.8.1.1 Objectives**

8232 The objectives of this sub-activity are to determine whether the formal security policy model of the
 8233 TSF clearly and consistently describes the rules and characteristics of the security policies and
 8234 whether this description corresponds with the description of security functions in the functional
 8235 specification.

8236 **11.8.1.2 Input**

8237 The evaluation evidence for this sub-activity is:

8238 e) the ST;

8239 the functional specification;

8240 formal security policy model (ADV_SPM.1.1D);

8241 formal proof of correspondence between the model and any formal functional specification
 8242 (ADV_SPM.1.3D);

8243 demonstration of correspondence between the model and the functional specification
 8244 (ADV_SPM.1.4D).

8245 **11.8.1.3 Application notes**

8246 This activity applies to cases where the developer has provided a formal security policy model of
 8247 the TOE.

8248 A formal TOE security policy model is a representation of the rules (synonymously termed
 8249 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
 8250 Their formal counterparts are called security properties and security features, respectively. The
 8251 representation includes but is not limited to algebraic specifications, finite state machines and logic
 8252 formalisms strong enough to formally infer the properties from the features. The formal TSP model
 8253 is accompanied by an informal interpretation explaining how the rules and characteristics are
 8254 mapped to the respective properties and features.

8255 The creation of a formal security policy model helps to identify and eliminate ambiguous,
 8256 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been built,
 8257 the formal model serves the evaluation effort by contributing to the evaluator's judgement of how
 8258 well the developer has understood the security functionality being implemented and whether
 8259 there are inconsistencies between the security requirements and the TOE design. The confidence in
 8260 the model is accompanied by a proof that it contains no inconsistencies.

8261 A formal security model is a precise formal presentation of the important aspects of security and
 8262 their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that defines
 8263 the TOE security policy and the set of practises (characteristics) that regulates how the TSF
 8264 manages, protects, and otherwise controls the system resources. The model includes the set of
 8265 restrictions and properties that specify how information and computing resources are prevented
 8266 from being used to violate the SFRs, accompanied by a persuasive set of engineering arguments
 8267 showing that these restrictions and properties play a key role in the enforcement of the SFRs. It
 8268 consists both of the formalisms that express the security functionality, as well as ancillary text to
 8269 explain the model and to provide it with context. The security behaviour of the TSF is modelled

- 8270 both in terms of external behaviour (i.e. how the TSF interacts with the rest of the TOE and with its
8271 operational environment), as well as its internal behaviour.
- 8272 The Security Policy Model of the TOE is informally abstracted from its realisation by considering
8273 the proposed security requirements of the ST. The informal abstraction is taken to be successful if
8274 the TOE's principles turn out to be enforced by its characteristics. The purpose of formal methods
8275 lies within the enhancement of the rigour of enforcement. Informal arguments are always prone to
8276 fallacies; especially if relationships among subjects, objects and operations get more and more
8277 involved. In order to minimise the risk of insecure state arrivals the rules and characteristics of the
8278 security policy model are mapped to respective properties and features within some formal system,
8279 whose rigour and strength can afterwards be used to obtain the security properties by means of
8280 theorems and formal proof.
- 8281 While the term "formal security policy model" is used in academic circles, the CC's approach has no
8282 fixed definition of "security"; it would equate to whatever SFRs are being claimed. Therefore, the
8283 formal security policy model is merely a formal representation of the set of SFRs being claimed.
- 8284 The term security policy has traditionally been associated with only access control policies,
8285 whether label-based (mandatory access control) or user-based (discretionary access control).
8286 However, a security policy is not limited to access control; there are also audit policies,
8287 identification policies, authentication policies, encryption policies, management policies, and any
8288 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
8289 contains an assignment for identifying these policies that are formally modelled.
- 8290 It is recognized that not all policies can be formally modelled for all TOEs. This is because either a
8291 given policy can not be formally modelled in the otherwise well suited framework, or because the
8292 nature of the TOE renders impossible the modelling of policies that would otherwise be possible to
8293 model.
- 8294 **11.8.1.4 Action ADV_SPM.1.1E**
- 8295 **ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory text as*
8296 *required, and identify the security policies of the TSF that are modelled.*
- 8297 **11.8.1.4.1 Work unit ADV_SPM.1-1**
- 8298 The evaluator **shall examine** the TOE security policy model to determine that it is written in a
8299 formal style.
- 8300 The evaluator identifies the formal framework upon which the TOE security policy model is based
8301 and ensures that it is founded on well established mathematical concepts. They also identify the
8302 security properties and features addressed in the application notes and ensure the formalization of
8303 at least one security policy.
- 8304 For guidance on formal methods refer to ISO/IEC 15408-3
- 8305 **11.8.1.4.2 Work unit ADV_SPM.1-2**
- 8306 The evaluator **shall examine** the TOE security policy model to determine that it contains all
8307 necessary informal explanatory text.
- 8308 Supporting narrative descriptions are necessary for all parts of the model (for example, to make
8309 clear the meaning of any formal notation and how they are used) including the security properties
8310 and features.

8311 **11.8.1.4.3 Work unit ADV_SPM.1-3**

8312 The evaluator *shall examine* the TOE security policy model to determine that all security policies
8313 of the TSF are identified that are modelled.

8314 The evaluator determines whether the SPM identifies the security policies for which a model is
8315 provided, identifying the relevant portions of the statement of SFRs that comprise each of the
8316 modelled policies.

8317 The evaluator determines whether the list of security policies identified by the SPM is consistent
8318 with the assignment of ADV_SPM.1.1D in the ST.

8319 The evaluator determines whether for each security policy identified by the SPM a model is in fact
8320 provided.

8321 **ADV_SPM.1.2C** *For all policies that are modelled, the model shall define security for the*
8322 *TOE and provide a formal proof that the TOE cannot reach a state that is*
8323 *not secure.*

8324 **11.8.1.4.4 Work unit ADV_SPM.1-4**

8325 The evaluator *shall examine* the principles and characteristics of the security policies to determine
8326 that the modelled security behaviour of the TOE is clearly articulated.

8327 The security policies are expressed in terms of security principles (rules) which are modelled by
8328 security properties and define the secure state of the TOE. For example, a model based on state
8329 transitions could describe the security policies in terms of principles of its states, identify its initial
8330 state, and define what it means to be a secure state.

8331 The evaluator determines that the security policies are reflected within their formal counterparts
8332 of the TSP model.

8333 The TOE security behaviour is expressed in terms of security characteristics (i.e. portions of TOE
8334 security functionality managing, protecting, and otherwise controlling the system resources
8335 including attributes and conditions of the TOE) which are modelled by security features. For
8336 example, a model based on state transitions could describe the characteristics as possible actions
8337 in each secure state in a level of detail sufficient to decide into which state the TOE will be
8338 transformed by that action.

8339 Together the security principles and characteristics describe the entire security posture of the TOE.

8340 In the context of a formal TOE security policy model the security behaviour is considered to be
8341 clearly articulated only if an adequate mapping from principles and characteristics to their
8342 respective formal counterparts properties and features has been given. The mapping is considered
8343 to be adequate if the level of abstraction from the TOE's realization is detailed enough to allow for
8344 correct identification of all security objectives and the relation to the security environment.

8345 The above condition for clear articulation is necessary but not sufficient. An informal
8346 interpretation of all formal concepts (including attributes, predicates and variables, if available)
8347 must be provided in order to make clear their intended meaning.

8348 **11.8.1.4.5 Work unit ADV_SPM.1-5**

8349 The evaluator *shall examine* the TOE security policy model rationale to determine that it formally
8350 proves that the security features enforce the security properties.

8351 To determine the enforcement, the evaluator considers the security properties and the security
 8352 features and verifies that the arguments used in the proof are valid. The proof of correspondence
 8353 between the security properties and the security features shall be formal.

8354 The validity of the security properties shall mean that the TOE is in a secure state. By this, the
 8355 evaluator confirms by means of the rationale that the TOE never reaches an insecure state.

8356 **11.8.1.4.6 Work unit ADV_SPM.1-6**

8357 The evaluator ***shall examine*** the TOE security policy model rationale to determine that it proves
 8358 the internal consistency of the TOE security policy model.

8359 The proof shall show the absence of contradictions within the TOE security policy model. In
 8360 determining the absence of contradictions, the evaluator verifies that the arguments used in the
 8361 proof are valid.

8362 Since the TOE security policy model is formal, the proof of its internal consistency shall be formal.
 8363 It is recognized that a complete formal proof of the internal consistency of the TOE security policy
 8364 model usually is not possible due to the fundamental nature of formal frameworks. Generally, it is
 8365 sufficient to generate evidence using formal proofs based on the specific TOE security policy model
 8366 that prove the internal consistency by means of a combination with generic arguments of the
 8367 formal framework.

8368 **ADV_SPM.1.3C** *The correspondence between the model and the functional specification*
 8369 *shall be at the correct level of formality.*

8370 **11.8.1.4.7 Work unit ADV_SPM.1-7**

8371 The evaluator ***shall examine*** the correspondence between the model and the functional
 8372 specification to determine that a semiformal demonstration of correspondence between the model
 8373 and any semiformal functional specification is provided.

8374 This work unit is only applicable to a semiformal presentation of the functional specification, which
 8375 is required by ADV_FSP.5.2C.

8376 A semiformal correspondence is one that results from a structured approach with a substantial
 8377 degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 8378 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 8379 terms, and so it provides less ambiguity than would exist in an informal correspondence.

8380 For guidance on semiformal methods refer to Annex 3.1.1 'Semiformal and formal methods'.

8381 **11.8.1.4.8 Work unit ADV_SPM.1-8**

8382 The evaluator ***shall examine*** the correspondence between the model and the functional
 8383 specification to determine that a formal proof of correspondence between the model and any
 8384 formal functional specification is provided.

8385 This work unit is only applicable to a formal presentation of the functional specification, which is
 8386 required by ADV_FSP.6.2D.

8387 There should be a formal proof of correspondence between the model and any formal functional
 8388 specification.

8389 The formal proof of correspondence removes all subjective interpretations of its terms by enlisting
 8390 well-established mathematical concepts to define the syntax and semantics of the formal notation
 8391 and uses rules that support logical reasoning. The security features within the TOE (which are

8392 identified in the formal TSP model) are expressed in a formal specification language and shown to
8393 be satisfied by the formal specification.

8394 For guidance on formal methods refer to ISO/IEC 15408-3.

8395 **ADV_SPM.1.4C** *The correspondence shall show that the functional specification is*
8396 *consistent and complete with respect to the model.*

8397 **11.8.1.4.9 Work unit ADV_SPM.1-9**

8398 The evaluator **shall examine** the correspondence to determine that the behaviour at the TSF
8399 interfaces (as articulated in the functional specification) is complete with respect to the behaviour
8400 modelled by the security features.

8401 The term “correspondence” here means both the formal proof of correspondence between the
8402 formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration of
8403 correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.

8404 In determining completeness of the correspondence, the evaluator considers the description of
8405 TSFI behaviour and maps adequate portions (characteristics) to corresponding features of the TSP
8406 model. The demonstration should show that all characteristics belonging to policies that are
8407 required to be modelled have an associated feature description in the TOE security policy model,
8408 and that each feature of the TSP model does occur in the mapping.

8409 Abstention from formally modelling TSFI behaviour always calls for justification on the developer’s
8410 side (also confer the application notes above).

8411 **11.8.1.4.10 Work unit ADV_SPM.1-10**

8412 The evaluator **shall examine** the correspondence to determine that the behaviour at the TSF
8413 interfaces (as articulated in the functional specification) is consistent with respect to the behaviour
8414 modelled by the security features.

8415 The term “correspondence” here means both the formal proof of correspondence between the
8416 formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration of
8417 correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

8418 The meaning of consistency reflects the conventional understanding in contrast to the internal
8419 consistency concept of work unit ADV_SPM.1-6.

8420 In determining consistency, the evaluator resumes the mapping of TSFI behaviour to security
8421 features established in the preceding work unit and verifies that the correspondence shows that
8422 each security feature of the TSP model accurately reflects the corresponding TSFI behaviour.

8423 For example, if TSFI behaviour dealt with access management on the granularity of single
8424 individuals, then a TSP model describing the security behaviour of the TOE in terms of groups of
8425 users would not be consistent. Likewise, if TSFI behaviour dealt with access management for
8426 groups of users, then a TSP model describing the security behaviour of the TOE in terms of
8427 individual users would also not be consistent.

8428 As another example, if remote untrusted users had to pass more stringent authentication
8429 procedures than administrators whose only point of access were within a physically-protected
8430 area, then this difference in authentication procedures had to be reflected in the security features.

8431 **11.9 TOE design (ADV_TDS)**

8432 **11.9.1 Evaluation of sub-activity (ADV_TDS.1)**

8433 **11.9.1.1 Input**

8434 The evaluation evidence for this sub-activity is:

- 8435 a) the ST;
- 8436 b) the functional specification;
- 8437 c) security architecture description;
- 8438 d) the TOE design.

8439 **11.9.1.2 Action ADV_TDS.1.1E**

8440 ISO/IEC 15408-3 ADV_TDS.1.1C: *The design shall describe the structure of the TOE in terms of*
8441 *subsystems.*

8442 **11.9.1.2.1 Work unit ADV_TDS.1-1**

8443 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is
8444 described in terms of subsystems.

8445 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the
8446 TOE will be used as input to work unit ADV_TDS.1-2, where the parts of the TOE that make up the
8447 TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

8448 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and
8449 modules) Depending upon the complexity of the TOE, its design may be described in terms of
8450 subsystems and modules, as described in ISO/IEC 15408-3 **Annex A.4, ADV_TDS: Subsystems and**
8451 **Modules**. At this level of assurance, the decomposition only need be at the “subsystem” level.

8452 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST,
8453 operator user guidance) to determine that the description of the TOE in such evidence is consistent
8454 with the description contained in the TOE design.

8455 ISO/IEC 15408-3 ADV_TDS.1.2C: *The design shall identify all subsystems of the TSF.*

8456 **11.9.1.2.2 Work unit ADV_TDS.1-2**

8457 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are
8458 identified.

8459 In work unit ADV_TDS.1-1 all of the subsystems of the TOE were identified, and a determination
8460 made that the non-TSF subsystems were correctly characterised. Building on that work, the
8461 subsystems that were not characterised as non-TSF subsystems should be precisely identified. The
8462 evaluator determines that, of the hardware and software installed and configured according to the
8463 Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one
8464 that is part of the TSF, or one that is not.

8465 ISO/IEC 15408-3 ADV_TDS.1.3C: *The design shall describe the behaviour of each SFR-supporting or*
8466 *SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.*

8467 **11.9.1.2.3 Work unit ADV_TDS.1-3**

8468 The evaluator **shall examine** the TOE design to determine that each SFR-supporting or SFR-non-
8469 interfering subsystem of the TSF is described such that the evaluator can determine that the
8470 subsystem is SFR-supporting or SFR-non-interfering.

8471 SFR-supporting and SFR-non-interfering subsystems do not need to be described in detail as to
8472 how they function in the system. However, the evaluator makes a determination, based on the
8473 evidence provided by the developer, that the subsystems that do not have high-level descriptions
8474 are SFR-supporting or SFR-non-interfering. Note that if the developer provides a uniform level of
8475 detailed documentation then this work unit will be largely satisfied, since the point of categorising
8476 the subsystems is to allow the developer to provide less information for SFR-supporting and SFR-
8477 non-interfering subsystems than for SFR-enforcing subsystems.

8478 An SFR-supporting subsystem is one that is depended on by an SFR-enforcing subsystem in order
8479 to implement an SFR, but does not play as direct a role as an SFR-enforcing subsystem. An SFR-
8480 non-interfering subsystem is one that is not depended upon, in either a supporting or enforcing
8481 role, to implement an SFR.

8482 ISO/IEC 15408-3 ADV_TDS.1.4C: *The design shall summarise the SFR-enforcing behaviour of the*
8483 *SFR-enforcing subsystems.*

8484 **11.9.1.2.4 Work unit ADV_TDS.1-4**

8485 The evaluator **shall examine** the TOE design to determine that it provides a complete, accurate,
8486 and high-level summary of the SFR-enforcing behaviour of the SFR-enforcing subsystems.

8487 The developer may designate subsystems as SFR-enforcing, SFR-supporting, and SFR non-
8488 interfering, but these “tags” are used only to describe the amount and type of information the
8489 developer must provide, and can be used to limit the amount of information the developer has to
8490 develop if their engineering process does not produce the documentation required. Whether the
8491 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to
8492 determine that the subsystems have the appropriate information for their role (SFR-enforcing, etc.)
8493 in the TOE, and to obtain the appropriate information from the developer should the developer fail
8494 to provide the required information for a particular subsystem.

8495 SFR-enforcing behaviour refers to how a subsystem provides the functionality that implements an
8496 SFR. The goal of evaluator's assessment is to give the evaluator with an understanding of the way
8497 each SFR-enforcing subsystem works. The information provided for the behaviour summary does
8498 not have to be as detailed as that provided by the behaviour description. For example, data
8499 structures or data items will likely not need to be described in detail. It is the evaluator's
8500 determination, however, with respect to what “high-level” means for a particular TOE, and the
8501 evaluator obtains enough information from the developer (even if it turns out to be equivalent to
8502 information provided for subsystem behaviour) to make a sound verdict for this work unit.

8503 The evaluator is cautioned, however, that “perfect” assurance is not a goal nor required by this
8504 work unit, so judgement will have to be exercised in determine the amount and composition of the
8505 evidence required to make a verdict on this work unit.

8506 To determine completeness and accuracy, the evaluator examines other information available (e.g.,
8507 functional specification, security architecture description). Summaries of functionality in these
8508 documents should be consistent with what is provided for evidence for this work unit.

8509 ISO/IEC 15408-3 ADV_TDS.1.5C: *The design shall provide a description of the interactions among*
8510 *SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other*
8511 *subsystems of the TSF.*

8512 **11.9.1.2.5 Work unit ADV_TDS.1-5**

8513 The evaluator ***shall examine*** the TOE design to determine that interactions between the
8514 subsystems of the TSF are described.

8515 The goal of describing the interactions between the SFR-enforcing subsystems and other
8516 subsystems is to help provide the reader a better understanding of how the TSF performs its
8517 functions. These interactions do not need to be characterised at the implementation level (e.g.,
8518 parameters passed from one routine in a subsystem to a routine in a different subsystem; global
8519 variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling
8520 subsystem), but the data elements identified for a particular subsystem that are going to be used by
8521 another subsystem need to be covered in this discussion. Any control relationships between
8522 subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the
8523 subsystem that actually implements these rules) should also be described.

8524 The evaluators need to use their own judgement in assessing the completeness of the description.
8525 If the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for
8526 instance, in examining the descriptions of subsystem behaviour) that do not appear to be described,
8527 the evaluator ensures that this information is provided by the developer. However, if the evaluator
8528 can determine that interactions among a particular set of subsystems, while incompletely
8529 described by the developer, will not aid in understanding the overall functionality nor security
8530 functionality provided by the TSF, then the evaluator may choose to consider the description
8531 sufficient, and not pursue completeness for its own sake.

8532 ISO/IEC 15408-3 ADV_TDS.1.6C: *The mapping shall demonstrate that all TSFIs trace to the*
8533 *behaviour described in the TOE design that they invoke.*

8534 **11.9.1.2.6 Work unit ADV_TDS.1-6**

8535 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate
8536 mapping from the TSFI described in the functional specification to the subsystems of the TSF
8537 described in the TOE design.

8538 The subsystems described in the TOE design provide a description of how the TSF works at a
8539 detailed level for SFR-enforcing portions of the TSF, and at a higher level for other portions of the
8540 TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the
8541 developer identifies the subsystem that is initially involved when an operation is requested at the
8542 TSFI, and identify the various subsystems that are primarily responsible for implementing the
8543 functionality. Note that a complete “call tree” for each TSFI is not required for this work unit.

8544 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at
8545 least one subsystem. The verification of accuracy is more complex.

8546 The first aspect of accuracy is that each TSFI is mapped to a subsystem at the TSF boundary. This
8547 determination can be made by reviewing the subsystem description and interactions, and from this
8548 information determining its place in the architecture. The next aspect of accuracy is that the
8549 mapping makes sense. For instance, mapping a TSFI dealing with access control to a subsystem
8550 that checks passwords is not accurate. The evaluator should again use judgement in making this
8551 determination. The goal is that this information aids the evaluator in understanding the system and
8552 implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the
8553 TSF. The bulk of the assessment of whether the SFRs are described accurately by the subsystems is
8554 performed in other work units.

8555 **11.9.1.3 Action ADV_TDS.1.2E**

8556 **11.9.1.3.1 Work unit ADV_TDS.1-7**

8557 The evaluator *shall examine* the TOE security functional requirements and the TOE design, to
8558 determine that all ST security functional requirements are covered by the TOE design.

8559 The evaluator may construct a map between the TOE security functional requirements and the TOE
8560 design. This map will likely be from a functional requirement to a set of subsystems. Note that this
8561 map may have to be at a level of detail below the component or even element level of the
8562 requirements, because of operations (assignments, refinements, selections) performed on the
8563 functional requirement by the ST author.

8564 For example, the **FDP_ACC.1 Subset access control** component contains an element with
8565 assignments. If the ST contained, for instance, ten rules in the **FDP_ACC.1 Subset access control**
8566 assignment, and these ten rules were implemented in specific places within fifteen modules, it
8567 would be inadequate for the evaluator to map **FDP_ACC.1 Subset access control** to one subsystem
8568 and claim the work unit had been completed. Instead, the evaluator would map **FDP_ACC.1 Subset**
8569 **access control** (rule 1) to subsystem A, behaviours x, y, and z; **FDP_ACC.1 Subset access control**
8570 (rule 2) to subsystem A, behaviours x, p, and q; etc.

8571 **11.9.1.3.2 Work unit ADV_TDS.1-8**

8572 The evaluator *shall examine* the TOE design to determine that it is an accurate instantiation of all
8573 security functional requirements.

8574 The evaluator ensures that each security requirement listed in the TOE security functional
8575 requirements subclause of the ST has a corresponding design description in the TOE design that
8576 accurately details how the TSF meets that requirement. This requires that the evaluator identify a
8577 collection of subsystems that are responsible for implementing a given functional requirement, and
8578 then examine those subsystems to understand how the requirement is implemented. Finally, the
8579 evaluator would assess whether the requirement was accurately implemented.

8580 As an example, if the ST requirements specified a role-based access control mechanism, the
8581 evaluator would first identify the subsystems that contribute to this mechanism's implementation.
8582 This could be done by in-depth knowledge or understanding of the TOE design or by work done in
8583 the previous work unit. Note that this trace is only to identify the subsystems, and is not the
8584 complete analysis.

8585 The next step would be to understand what mechanism the subsystems implemented. For instance,
8586 if the design described an implementation of access control based on UNIX-style protection bits,
8587 the design would not be an accurate instantiation of those access control requirements present in
8588 the ST example used above. If the evaluator could not determine that the mechanism was
8589 accurately implemented because of a lack of detail, the evaluator would have to assess whether all
8590 of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for
8591 those subsystems.

8592 **11.9.2 Evaluation of sub-activity (ADV_TDS.2)**

8593 **11.9.2.1 Input**

8594 The evaluation evidence for this sub-activity is:

- 8595 a) the ST;
- 8596 b) the functional specification;
- 8597 c) security architecture description;

- 8598 d) the TOE design.
- 8599 **11.9.2.2 Action ADV_TDS.2.1E**
- 8600 ISO/IEC 15408-3 ADV_TDS.2.1C: *The design shall describe the structure of the TOE in terms of*
8601 *subsystems.*
- 8602 **11.9.2.2.1 Work unit ADV_TDS.2-1**
- 8603 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is
8604 described in terms of subsystems.
- 8605 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the
8606 TOE will be used as input to work unit ADV_TDS.2-2, where the parts of the TOE that make up the
8607 TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.
- 8608 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and
8609 modules) Depending upon the complexity of the TOE, its design may be described in terms of
8610 subsystems and modules, as described in ISO/IEC 15408-3 **Annex A.4, ADV_TDS: Subsystems and**
8611 **Modules**. At this level of assurance, the decomposition only need be at the “subsystem” level.
- 8612 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST,
8613 operator user guidance) to determine that the description of the TOE in such evidence is consistent
8614 with the description contained in the TOE design.
- 8615 ISO/IEC 15408-3 ADV_TDS.2.2C: *The design shall identify all subsystems of the TSF.*
- 8616 **11.9.2.2.2 Work unit ADV_TDS.2-2**
- 8617 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are
8618 identified.
- 8619 In work unit ADV_TDS.2-1 all of the subsystems of the TOE were identified, and a determination
8620 made that the non-TSF subsystems were correctly characterised. Building on that work, the
8621 subsystems that were not characterised as non-TSF subsystems should be precisely identified. The
8622 evaluator determines that, of the hardware and software installed and configured according to the
8623 Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one
8624 that is part of the TSF, or one that is not.
- 8625 ISO/IEC 15408-3 ADV_TDS.2.3C: *The design shall describe the behaviour of each SFR non-interfering*
8626 *subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.*
- 8627 **11.9.2.2.3 Work unit ADV_TDS.2-3**
- 8628 The evaluator ***shall examine*** the TOE design to determine that each SFR-non-interfering
8629 subsystem of the TSF is described such that the evaluator can determine that the subsystem is SFR-
8630 non-interfering.
- 8631 SFR-non-interfering subsystems do not need to be described in detail as to how they function in
8632 the system. However, the evaluator makes a determination, based on the evidence provided by the
8633 developer, that the subsystems that do not have detailed descriptions are SFR-non-interfering.
8634 Note that if the developer provides a uniform level of detailed documentation then this work unit
8635 will be largely satisfied, since the point of categorising the subsystems is to allow the developer to
8636 provide less information for SFR-non-interfering subsystems than for SFR-enforcing and SFR-
8637 supporting subsystems.
- 8638 An SFR-non-interfering subsystem is one on which the SFR-enforcing and SFR-supporting
8639 subsystems have no dependence; that is, they play no role in implementing SFR functionality.

8640 ISO/IEC 15408-3 ADV_TDS.2.4C: *The design shall describe the SFR-enforcing behaviour of the SFR-*
8641 *enforcing subsystems.*

8642 11.9.2.2.4 Work unit ADV_TDS.2-4

8643 The evaluator ***shall examine*** the TOE design to determine that it provides a complete, accurate,
8644 and detailed description of the SFR-enforcing behaviour of the SFR-enforcing subsystems.

8645 The developer may designate subsystems as SFR-enforcing, SFR-supporting, and SFR non-
8646 interfering, but these “tags” are used only to describe the amount and type of information the
8647 developer must provide, and can be used to limit the amount of information the developer has to
8648 develop if their engineering process does not produce the documentation required. Whether the
8649 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to
8650 determine that the subsystems have the appropriate information for their role (SFR-enforcing, etc.)
8651 in the TOE, and to obtain the appropriate information from the developer should the developer fail
8652 to provide the required information for a particular subsystem.

8653 SFR-enforcing behaviour refers to *how* a subsystem provides the functionality that implements an
8654 SFR. While not at the level of an algorithmic description, a detailed description of behaviour
8655 typically discusses how the functionality is provided in terms of what key data and data structures
8656 are, what control relationships exist within a subsystem, and how these elements work together to
8657 provide the SFR-enforcing behaviour. Such a description also references SFR-supporting behaviour,
8658 which the evaluator should consider in performing subsequent work units.

8659 To determine completeness and accuracy, the evaluator examines other information available (e.g.,
8660 functional specification, security architecture description). Descriptions of functionality in these
8661 documents should be consistent with what is provided for evidence for this work unit.

8662 ISO/IEC 15408-3 ADV_TDS.2.5C: *The design shall summarise the SFR-supporting and SFR-non-*
8663 *interfering behaviour of the SFR-enforcing subsystems.*

8664 11.9.2.2.5 Work unit ADV_TDS.2-5

8665 The evaluator ***shall examine*** the TOE design to determine that it provides a complete and accurate
8666 high-level summary of the SFR-supporting and SFR-non-interfering behaviour of the SFR-enforcing
8667 subsystems.

8668 The developer may designate subsystems as SFR-enforcing, SFR-supporting, and SFR non-
8669 interfering, but these “tags” are used only to describe the amount and type of information the
8670 developer must provide, and can be used to limit the amount of information the developer has to
8671 develop if their engineering process does not produce the documentation required. Whether the
8672 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to
8673 determine that the subsystems have the appropriate information for their role (SFR-enforcing, etc.)
8674 in the TOE, and to obtain the appropriate information from the developer should the developer fail
8675 to provide the required information for a particular subsystem.

8676 In contrast to the previous work unit, this work unit calls for the evaluator to assess the
8677 information provided for SFR-enforcing subsystems that is SFR-supporting or SFR-non-interfering.
8678 The goal of this assessment is two-fold. First, it should provide the evaluator greater understanding
8679 of the way each subsystem works. Second, this assessment will help the evaluator to determine
8680 that all SFR-enforcing behaviour exhibited by a SFR-enforcing subsystem has been described.
8681 Unlike the previous work unit, the information provided for the SFR-supporting or SFR-non-
8682 interfering behaviour does not have to be as detailed as that provided by the SFR-enforcing
8683 behaviour. For example, data structures or data items that do not pertain to SFR-enforcing
8684 functionality will likely not need to be described in detail, if at all. It is the evaluator's
8685 determination, however, with respect to what “high-level” means for a particular TOE, and the
8686 evaluator obtains enough information from the developer (even if it turns out to be equivalent to

8687 information provided for the parts of the subsystem that are SFR-enforcing) to make a sound
8688 verdict for this work unit.

8689 The evaluator is cautioned, however, that “perfect” assurance is not a goal nor required by this
8690 work unit, so judgement will have to be exercised in determine the amount and composition of the
8691 evidence required to make a verdict on this work unit.

8692 To determine completeness and accuracy, the evaluator examines other information available (e.g.,
8693 functional specification, security architecture description). Summaries of functionality in these
8694 documents should be consistent with what is provided for evidence for this work unit. In particular,
8695 the functional specification should be used to determine that the behaviour required to implement
8696 the TSF Interfaces described by the functional specification are completely described by the
8697 subsystem, since the behaviour will either be SFR-enforcing, SFR-supporting or SFR-non-
8698 interfering.

8699 ISO/IEC 15408-3 ADV_TDS.2.6C: *The design shall summarise the behaviour of the SFR-supporting*
8700 *subsystems.*

8701 **11.9.2.2.6 Work unit ADV_TDS.2-6**

8702 The evaluator **shall examine** the TOE design to determine that it provides a complete and accurate
8703 high-level summary of the behaviour of the SFR-supporting subsystems.

8704 The developer may designate subsystems as SFR-enforcing, SFR-supporting, and SFR non-
8705 interfering, but these “tags” are used only to describe the amount and type of information the
8706 developer must provide, and can be used to limit the amount of information the developer has to
8707 develop if their engineering process does not produce the documentation required. Whether the
8708 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to
8709 determine that the subsystems have the appropriate information for their role (SFR-enforcing, etc.)
8710 in the TOE, and to obtain the appropriate information from the developer should the developer fail
8711 to provide the required information for a particular subsystem.

8712 In contrast to the previous two work units, this work unit calls for the developer to provide (and
8713 the evaluator to assess) information about SFR supporting subsystems. Such subsystems should be
8714 referenced by the descriptions of the SFR-enforcing subsystems, as well as by the descriptions of
8715 interactions in work unit ADV_TDS.2-7. The goal of evaluator's assessment, like that for the
8716 previous work unit, is two-fold. First, it should provide the evaluator with an understanding of the
8717 way each SFR-supporting subsystem works. Second, the evaluator determines that the behaviour is
8718 summarized in enough detail so that the way in which the subsystem supports the SFR-enforcing
8719 behaviour is clear, and that the behaviour is not itself SFR-enforcing. The information provided for
8720 SFR-supporting subsystem's behaviour does not have to be as detailed as that provided by the SFR-
8721 enforcing behaviour. For example, data structures or data items that do not pertain to SFR-
8722 enforcing functionality will likely not need to be described in detail, if at all. It is the evaluator's
8723 determination, however, with respect to what “high-level” means for a particular TOE, and the
8724 evaluator obtains enough information from the developer (even if it turns out to be equivalent to
8725 information provided for the parts of the subsystem that are SFR-enforcing) to make a sound
8726 verdict for this work unit.

8727 The evaluator is cautioned, however, that “perfect” assurance is not a goal nor required by this
8728 work unit, so judgement will have to be exercised in determine the amount and composition of the
8729 evidence required to make a verdict on this work unit.

8730 To determine completeness and accuracy, the evaluator examines other information available (e.g.,
8731 functional specification, security architecture description). Summaries of functionality in these
8732 documents should be consistent with what is provided for evidence for this work unit.

8733 ISO/IEC 15408-3 ADV_TDS.2.7C: *The design shall provide a description of the interactions among all*
8734 *subsystems of the TSF.*

8735 **11.9.2.2.7 Work unit ADV_TDS.2-7**

8736 The evaluator **shall examine** the TOE design to determine that interactions between the
8737 subsystems of the TSF are described.

8738 The goal of describing the interactions between the subsystems is to help provide the reader a
8739 better understanding of how the TSF performs its functions. These interactions do not need to be
8740 characterised at the implementation level (e.g., parameters passed from one routine in a subsystem
8741 to a routine in a different subsystem; global variables; hardware signals (e.g., interrupts) from a
8742 hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a
8743 particular subsystem that are going to be used by another subsystem need to be covered in this
8744 discussion. Any control relationships between subsystems (e.g., a subsystem responsible for
8745 configuring a rule base for a firewall system and the subsystem that actually implements these
8746 rules) should also be described.

8747 It should be noted while the developer should characterise all interactions between subsystems,
8748 the evaluators need to use their own judgement in assessing the completeness of the description. If
8749 the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for
8750 instance, in examining the descriptions of subsystem behaviour) that do not appear to be described,
8751 the evaluator ensures that this information is provided by the developer. However, if the evaluator
8752 can determine that interactions among a particular set of subsystems, while incompletely
8753 described by the developer, will not aid in understanding the overall functionality nor security
8754 functionality provided by the TSF, then the evaluator may choose to consider the description
8755 sufficient, and not pursue completeness for its own sake.

8756 ISO/IEC 15408-3 ADV_TDS.2.8C: *The mapping shall demonstrate that all TSFIs trace to the*
8757 *behaviour described in the TOE design that they invoke.*

8758 **11.9.2.2.8 Work unit ADV_TDS.2-8**

8759 The evaluator **shall examine** the TOE design to determine that it contains a complete and accurate
8760 mapping from the TSFI described in the functional specification to the subsystems of the TSF
8761 described in the TOE design.

8762 The subsystems described in the TOE design provide a description of how the TSF works at a
8763 detailed level for SFR-enforcing portions of the TSF, and at a higher level for other portions of the
8764 TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the
8765 developer identifies the subsystem that is initially involved when an operation is requested at the
8766 TSFI, and identify the various subsystems that are primarily responsible for implementing the
8767 functionality. Note that a complete “call tree” for each TSFI is not required for this work unit.

8768 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at
8769 least one subsystem. The verification of accuracy is more complex.

8770 The first aspect of accuracy is that each TSFI is mapped to a subsystem at the TSF boundary. This
8771 determination can be made by reviewing the subsystem description and interactions, and from this
8772 information determining its place in the architecture. The next aspect of accuracy is that the
8773 mapping makes sense. For instance, mapping a TSFI dealing with access control to a subsystem
8774 that checks passwords is not accurate. The evaluator should again use judgement in making this
8775 determination. The goal is that this information aids the evaluator in understanding the system and
8776 implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the
8777 TSF. The bulk of the assessment of whether the SFRs are described accurately by the subsystems is
8778 performed in other work units.

8779 **11.9.2.3 Action ADV_TDS.2.2E**8780 **11.9.2.3.1 Work unit ADV_TDS.2-9**

8781 The evaluator *shall examine* the TOE security functional requirements and the TOE design, to
 8782 determine that all ST security functional requirements are covered by the TOE design.

8783 The evaluator may construct a map between the TOE security functional requirements and the TOE
 8784 design. This map will likely be from a functional requirement to a set of subsystems. Note that this
 8785 map may have to be at a level of detail below the component or even element level of the
 8786 requirements, because of operations (assignments, refinements, selections) performed on the
 8787 functional requirement by the ST author.

8788 For example, the **FDP_ACC.1 Subset access control** component contains an element with
 8789 assignments. If the ST contained, for instance, ten rules in the **FDP_ACC.1 Subset access control**
 8790 assignment, and these ten rules were implemented in specific places within fifteen modules, it
 8791 would be inadequate for the evaluator to map **FDP_ACC.1 Subset access control** to one subsystem
 8792 and claim the work unit had been completed. Instead, the evaluator would map **FDP_ACC.1 Subset**
 8793 **access control** (rule 1) to subsystem A, behaviours x, y, and z; **FDP_ACC.1 Subset access control**
 8794 (rule 2) to subsystem A, behaviours x, p, and q; etc.

8795 **11.9.2.3.2 Work unit ADV_TDS.2-10**

8796 The evaluator *shall examine* the TOE design to determine that it is an accurate instantiation of all
 8797 security functional requirements.

8798 The evaluator ensures that each security requirement listed in the TOE security functional
 8799 requirements subclause of the ST has a corresponding design description in the TOE design that
 8800 accurately details how the TSF meets that requirement. This requires that the evaluator identify a
 8801 collection of subsystems that are responsible for implementing a given functional requirement, and
 8802 then examine those subsystems to understand how the requirement is implemented. Finally, the
 8803 evaluator would assess whether the requirement was accurately implemented.

8804 As an example, if the ST requirements specified a role-based access control mechanism, the
 8805 evaluator would first identify the subsystems that contribute to this mechanism's implementation.
 8806 This could be done by in-depth knowledge or understanding of the TOE design or by work done in
 8807 the previous work unit. Note that this trace is only to identify the subsystems, and is not the
 8808 complete analysis.

8809 The next step would be to understand what mechanism the subsystems implemented. For instance,
 8810 if the design described an implementation of access control based on UNIX-style protection bits,
 8811 the design would not be an accurate instantiation of those access control requirements present in
 8812 the ST example used above. If the evaluator could not determine that the mechanism was
 8813 accurately implemented because of a lack of detail, the evaluator would have to assess whether all
 8814 of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for
 8815 those subsystems.

8816 **11.9.3 Evaluation of sub-activity (ADV_TDS.3)**8817 **11.9.3.1 Objectives**

8818 The objective of this sub-activity is to determine whether the TOE design provides a description of
 8819 the TOE in terms of subsystems sufficient to determine the TSF boundary, and provides a
 8820 description of the TSF internals in terms of modules (and optionally higher-level abstractions). It
 8821 provides a detailed description of the SFR-enforcing modules and enough information about the
 8822 SFR-supporting and SFR-non-interfering modules for the evaluator to determine that the SFRs are
 8823 completely and accurately implemented; as such, the TOE design provides an explanation of the
 8824 implementation representation.

8825 **11.9.3.2 Input**

8826 The evaluation evidence for this sub-activity is:

- 8827 a) the ST;
- 8828 b) the functional specification;
- 8829 c) security architecture description;
- 8830 d) the TOE design.

8831 **11.9.3.3 Application notes**

8832 There are three types of activity that the evaluator must undertake with respect to the TOE design.
8833 First, the evaluator determines that the TSF boundary has been adequately described. Second, the
8834 evaluator determines that the developer has provided documentation that conforms to the content
8835 and presentation requirements for this subsystem, and that is consistent with other documentation
8836 provided for the TOE. Finally, the evaluator must analyse the design information provided for the
8837 SFR-enforcing modules (at a detailed level) and the SFR-supporting and SFR-non-interfering
8838 modules (at a less detailed level) to understand how the system is implemented, and with that
8839 knowledge ensure that the TSFI in the functional specification are adequately described, and that
8840 the test information adequately tests the TSF (done in the Class ATE: Tests work units).

8841 It is important to note that while the developer is obligated to provide a complete description of
8842 the TSF (although SFR-enforcing modules will have more detail than the SFR-supporting or SFR-
8843 non-interfering modules), the evaluator is expected to use their judgement in performing their
8844 analysis. While the evaluator is expected to look at every module, the detail to which they examine
8845 each module may vary. The evaluator analyses each module in order to gain enough understanding
8846 to determine the effect of the functionality of the module on the security of the system, and the
8847 depth to which they need to analyse the module may vary depending on the module's role in the
8848 system. An important aspect of this analysis is that the evaluator should use the other
8849 documentation provided (TSS, functional specification, security architecture description, and the
8850 TSF internal document) in order to determine that the functionality that is described is correct, and
8851 that the implicit designation of SFR-supporting or SFR-non-interfering modules (see below) is
8852 supported by their role in the system architecture.

8853 The developer may designate modules as SFR-enforcing, SFR-supporting, and SFR non-interfering,
8854 but these "tags" are used only to describe the amount and type of information the developer must
8855 provide, and can be used to limit the amount of information the developer has to develop if their
8856 engineering process does not produce the documentation required. Whether the modules have
8857 been categorised by the developer or not, it is the evaluator's responsibility to determine that the
8858 modules have the appropriate information for their role (SFR-enforcing, etc.) in the TOE, and to
8859 obtain the appropriate information from the developer should the developer fail to provide the
8860 required information for a particular module.

8861 **11.9.3.4 Action ADV_TDS.3.1E**

8862 ISO/IEC 15408-3 ADV_TDS.3.1C: *The design shall describe the structure of the TOE in terms of*
8863 *subsystems.*

8864 **11.9.3.4.1 Work unit ADV_TDS.3-1**

8865 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is
8866 described in terms of subsystems.

8867 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the
 8868 TOE will be used as input to work unit ADV_TDS.3-2, where the parts of the TOE that make up the
 8869 TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

8870 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and
 8871 modules). Depending upon the complexity of the TOE, its design may be described in terms of
 8872 subsystems and modules, as described in ISO/IEC 15408-3 **Annex A.4, ADV_TDS: Subsystems and**
 8873 **Modules**. For a very simple TOE that can be described solely at the “module” level (see ADV_TDS.3-
 8874 2), this work unit is not applicable and therefore considered to be satisfied.

8875 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST,
 8876 operator user guidance) to determine that the description of the TOE in such evidence is consistent
 8877 with the description contained in the TOE design.

8878 ISO/IEC 15408-3 ADV_TDS.3.2C: *The design shall describe the TSF in terms of modules.*

8879 **11.9.3.4.2 Work unit ADV_TDS.3-2**

8880 The evaluator **shall examine** the TOE design to determine that the entire TSF is described in terms
 8881 of modules.

8882 The evaluator will examine the modules for specific properties in other work units; in this work
 8883 unit the evaluator determines that the modular description covers the entire TSF, and not just a
 8884 portion of the TSF. The evaluator uses other evidence provided for the evaluation (e.g., functional
 8885 specification, security architecture description) in making this determination. For example, if the
 8886 functional specification contains interfaces to functionality that does not appear to be described in
 8887 the TOE design description, it may be the case that a portion of the TSF has not been included
 8888 appropriately. Making this determination will likely be an iterative process, where as more analysis
 8889 is done on the other evidence, more confidence can be gained with respect to the completeness of
 8890 the documentation.

8891 Unlike subsystems, modules describe the implementation in a level of detail that can serve as a
 8892 guide to reviewing the implementation representation. A description of a module should be such
 8893 that one could create an implementation of the module from the description, and the resulting
 8894 implementation would be 1) identical to the actual TSF implementation in terms of the interfaces
 8895 presented, 2) identical in the use of interfaces that are mentioned in the design, and 3) functionally
 8896 equivalent to the description of the purpose of the TSF module. For instance, RFC 793 provides a
 8897 high-level description of the TCP protocol. It is necessarily implementation independent. While it
 8898 provides a wealth of detail, it is **not** a suitable design description because it is not specific to an
 8899 implementation. An actual implementation can add to the protocol specified in the RFC, and
 8900 implementation choices (for instance, the use of global data vs. local data in various parts of the
 8901 implementation) may have an impact on the analysis that is performed. The design description of
 8902 the TCP module would list the interfaces presented by the implementation (rather than just those
 8903 defined in RFC 793), as well as an algorithm description of the processing associated with the
 8904 modules implementing TCP (assuming it was part of the TSF).

8905 ISO/IEC 15408-3 ADV_TDS.3.3C: *The design shall identify all subsystems of the TSF.*

8906 **11.9.3.4.3 Work unit ADV_TDS.3-3**

8907 The evaluator **shall examine** the TOE design to determine that all subsystems of the TSF are
 8908 identified.

8909 If the design is presented solely in terms of modules, then subsystems in these requirements are
 8910 equivalent to modules and the activity should be performed at the module level.

8911 In work unit ADV_TDS.3-1 all of the subsystems of the TOE were identified, and a determination
 8912 made that the non-TSF subsystems were correctly characterised. Building on that work, the

8913 subsystems that were not characterised as non-TSF subsystems should be precisely identified. The
8914 evaluator determines that, of the hardware and software installed and configured according to the
8915 Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one
8916 that is part of the TSF, or one that is not.

8917 ISO/IEC 15408-3 ADV_TDS.3.4C: *The design shall provide a description of each subsystem of the TSF.*

8918 **11.9.3.4.4 Work unit ADV_TDS.3-4**

8919 The evaluator ***shall examine*** the TOE design to determine that each subsystem of the TSF
8920 describes its role in the enforcement of SFRs described in the ST.

8921 If the design is presented solely in terms of modules, then this work unit will be considered
8922 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
8923 evaluator is necessary in this case.

8924 On systems that are complex enough to warrant a subsystem-level description of the TSF in
8925 addition to the modular description, the goal of the subsystem-level description is to give the
8926 evaluator context for the modular description that follows. Therefore, the evaluator ensures that
8927 the subsystem-level description contains a description of how the security functional requirements
8928 are achieved in the design, but at a level of abstraction above the modular description. This
8929 description should discuss the mechanisms used at a level that is aligned with the module
8930 description; this will provide the evaluators the road map needed to intelligently assess the
8931 information contained in the module description. A well-written set of subsystem descriptions will
8932 help guide the evaluator in determining the modules that are most important to examine, thus
8933 focusing the evaluation activity on the portions of the TSF that have the most relevance with
8934 respect to the enforcement of the SFRs.

8935 The evaluator ensures that all subsystems of the TSF have a description. While the description
8936 should focus on the role that the subsystem plays in enforcing or supporting the implementation of
8937 the SFRs, enough information must be present so that a context for understanding the SFR-related
8938 functionality is provided.

8939 **11.9.3.4.5 Work unit ADV_TDS.3-5**

8940 The evaluator ***shall examine*** the TOE design to determine that each SFR-non-interfering
8941 subsystem of the TSF is described such that the evaluator can determine that the subsystem is SFR-
8942 non-interfering.

8943 If the design is presented solely in terms of modules, then this work unit will be considered
8944 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
8945 evaluator is necessary in this case.

8946 An SFR-non-interfering subsystem is one on which the SFR-enforcing and SFR-supporting
8947 subsystems have no dependence; that is, they play no role in implementing SFR functionality.

8948 The evaluator ensures that all subsystems of the TSF have a description. While the description
8949 should focus on the role that the subsystem do not plays in enforcing or supporting the
8950 implementation of the SFRs, enough information must be present so that a context for
8951 understanding the SFR-non-interfering functionality is provided.

8952 ISO/IEC 15408-3 ADV_TDS.3.5C: *The design shall provide a description of the interactions among all*
8953 *subsystems of the TSF.*

8954 **11.9.3.4.6 Work unit ADV_TDS.3-6**

8955 The evaluator ***shall examine*** the TOE design to determine that interactions between the
8956 subsystems of the TSF are described.

8957 If the design is presented solely in terms of modules, then this work unit will be considered
 8958 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
 8959 evaluator is necessary in this case.

8960 On systems that are complex enough to warrant a subsystem-level description of the TSF in
 8961 addition to the modular description, the goal of describing the interactions between the
 8962 subsystems is to help provide the reader a better understanding of how the TSF performs its
 8963 functions. These interactions do not need to be characterised at the implementation level (e.g.,
 8964 parameters passed from one routine in a subsystem to a routine in a different subsystem; global
 8965 variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling
 8966 subsystem), but the data elements identified for a particular subsystem that are going to be used by
 8967 another subsystem should be covered in this discussion. Any control relationships between
 8968 subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the
 8969 subsystem that actually implements these rules) should also be described.

8970 It should be noted while the developer should characterise all interactions between subsystems,
 8971 the evaluators need to use their own judgement in assessing the completeness of the description. If
 8972 the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for
 8973 instance, in examining the module-level documentation) that do not appear to be described, the
 8974 evaluator ensures that this information is provided by the developer. However, if the evaluator can
 8975 determine that interactions among a particular set of subsystems, while incompletely described by
 8976 the developer, and a complete description will not aid in understanding the overall functionality
 8977 nor security functionality provided by the TSF, then the evaluator may choose to consider the
 8978 description sufficient, and not pursue completeness for its own sake.

8979 ISO/IEC 15408-3 ADV_TDS.3.6C: *The design shall provide a mapping from the subsystems of the TSF*
 8980 *to the modules of the TSF.*

8981 **11.9.3.4.7 Work unit ADV_TDS.3-7**

8982 The evaluator ***shall examine*** the TOE design to determine that the mapping between the
 8983 subsystems of the TSF and the modules of the TSF is complete.

8984 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

8985 For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition
 8986 to the modular description, the developer provides a simple mapping showing how the modules of
 8987 the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their
 8988 module-level assessment. To determine completeness, the evaluator examines each mapping and
 8989 determines that all subsystems map to at least one module, and that all modules map to exactly one
 8990 subsystem.

8991 **11.9.3.4.8 Work unit ADV_TDS.3-8**

8992 The evaluator ***shall examine*** the TOE design to determine that the mapping between the
 8993 subsystems of the TSF and the modules of the TSF is accurate.

8994 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

8995 For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition
 8996 to the modular description, the developer provides a simple mapping showing how the modules of
 8997 the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their
 8998 module-level assessment. The evaluator may choose to check the accuracy of the mapping in
 8999 conjunction with performing other work units. An “inaccurate” mapping is one where the module
 9000 is mistakenly associated with a subsystem where its functions are not used within the subsystem.
 9001 Because the mapping is intended to be a guide supporting more detailed analysis, the evaluator is
 9002 cautioned to apply appropriate effort to this work unit. Expending extensive evaluator resources
 9003 verifying the accuracy of the mapping is not necessary. Inaccuracies that lead to mis-

9004 understandings related to the design that are uncovered as part of this or other work units are the
9005 ones that should be associated with this work unit and corrected.

9006 ISO/IEC 15408-3 ADV_TDS.3.7C: *The design shall describe each SFR-enforcing module in terms of its*
9007 *purpose and relationship with other modules.*

9008 **11.9.3.4.9 Work unit ADV_TDS.3-9**

9009 The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of
9010 each SFR-enforcing module and relationship with other modules is complete and accurate.

9011 The developer may designate modules as SFR-enforcing, SFR-supporting, and SFR non-interfering,
9012 but these “tags” are used only to describe the amount and type of information the developer must
9013 provide, and can be used to limit the amount of information the developer has to develop if their
9014 engineering process does not produce the documentation required. Whether the modules have
9015 been categorised by the developer or not, it is the evaluator's responsibility to determine that the
9016 modules have the appropriate information for their role (SFR-enforcing, etc.) in the TOE, and to
9017 obtain the appropriate information from the developer should the developer fail to provide the
9018 required information for a particular module.

9019 The purpose of a module provides a description indicating what function the module is fulfilling. A
9020 word of caution to evaluator is in order. The focus of this work unit should be to provide the
9021 evaluator an understanding of how the module works so that determinations can be made about
9022 the soundness of the implementation of the SFRs, as well as to support architectural analysis
9023 performed for ADV_ARC component. As long as the evaluator has a sound understanding of the
9024 module's operation, and its relationship to other modules and the TOE as a whole, the evaluator
9025 should consider the objective of the work achieved and not engage in a documentation exercise for
9026 the developer (by requiring, for example, a complete algorithmic description for a self-evident
9027 implementation representation).

9028 Because the modules are at such a low level, it may be difficult determine completeness and
9029 accuracy impacts from other documentation, such as operational user guidance, the functional
9030 specification, the TSF internals, or the security architecture description. However, the evaluator
9031 uses the information present in those documents to the extent possible to help ensure that the
9032 purpose is accurately and completely described. This analysis can be aided by the analysis
9033 performed for the work units for the **ADV_TDS.3.10C** element, which maps the TSFI in the
9034 functional specification to the modules of the TSF.

9035 ISO/IEC 15408-3 ADV_TDS.3.8C: *The design shall describe each SFR-enforcing module in terms of its*
9036 *SFR-related interfaces, return values from those interfaces, interaction with other modules and called*
9037 *SFR-related interfaces to other SFR-enforcing modules.*

9038 **11.9.3.4.10 Work unit ADV_TDS.3-10**

9039 The evaluator ***shall examine*** the TOE design to determine that the description of the interfaces
9040 presented by each SFR-enforcing module contain an accurate and complete description of the SFR-
9041 related parameters, the invocation conventions for each interface, and any values returned directly
9042 by the interface.

9043 The SFR-related interfaces of a module are those interfaces used by other modules as a means to
9044 invoke the SFR-related operations provided, and to provide inputs to or receive outputs from the
9045 module. The purpose in the specification of these interfaces is to permit the exercise of them
9046 during testing. Inter-module interfaces that are not SFR-related need not be specified or described,
9047 since they are not a factor in testing. Likewise, other internal interfaces that are not a factor in
9048 traversing SFR-related paths of execution (such as those internal paths that are fixed) need not be
9049 specified or described, since they are not a factor in testing.

9050 SFR-related interfaces are described in terms of how they are invoked, and any values that are
 9051 returned. This description would include a list of SFR-related parameters, and descriptions of these
 9052 parameters. Note that global data would also be considered parameters if used by the module
 9053 (either as inputs or outputs) when invoked. If a parameter were expected to take on a set of values
 9054 (e.g., a “flag” parameter), the complete set of values the parameter could take on that would have
 9055 an effect on module processing would be specified. Likewise, parameters representing data
 9056 structures are described such that each field of the data structure is identified and described. Note
 9057 that different programming languages may have additional “interfaces” that would be non-obvious;
 9058 an example would be operator/function overloading in C++. This “implicit interface” in the class
 9059 description would also be described as part of the low-level TOE design. Note that although a
 9060 module could present only one interface, it is more common that a module presents a small set of
 9061 related interfaces.

9062 In terms of the assessment of parameters (inputs and outputs) to a module, any use of global data
 9063 must also be considered. A module “uses” global data if it either reads or writes the data. In order
 9064 to assure the description of such parameters (if used) is complete, the evaluator uses other
 9065 information provided about the module in the TOE design (interfaces, algorithmic description, etc.),
 9066 as well as the description of the particular set of global data assessed in work unit ADV_TDS.3-10.
 9067 For instance, the evaluator could first determine the processing the module performs by examining
 9068 its function and interfaces presented (particularly the parameters of the interfaces). They could
 9069 then check to see if the processing appears to “touch” any of the global data areas identified in the
 9070 TOE design. The evaluator then determines that, for each global data area that appears to be
 9071 “touched”, that global data area is listed as a means of input or output by the module the evaluator
 9072 is examining.

9073 Invocation conventions are a programming-reference-type description that one could use to
 9074 correctly invoke a module’s interface if one were writing a program to make use of the module’s
 9075 functionality through that interface. This includes necessary inputs and outputs, including any set-
 9076 up that may need to be performed with respect to global variables.

9077 Values returned through the interface refer to values that are either passed through parameters or
 9078 messages; values that the function call itself returns in the style of a “C” program function call; or
 9079 values passed through global means (such as certain error routines in *ix-style operating systems).

9080 In order to assure the description is complete, the evaluator uses other information provided about
 9081 the module in the TOE design (e.g., algorithmic description, global data used) to ensure that it
 9082 appears all data necessary for performing the functions of the module is presented to the module,
 9083 and that any values that other modules expect the module under examination to provide are
 9084 identified as being returned by the module. The evaluator determines accuracy by ensuring that
 9085 the description of the processing matches the information listed as being passed to or from an
 9086 interface.

9087 ISO/IEC 15408-3 ADV_TDS.3.9C: *The design shall describe each SFR-supporting or SFR-non-*
 9088 *interfering module in terms of its purpose and interaction with other modules.*

9089 **11.9.3.4.11 Work unit ADV_TDS.3-11**

9090 The evaluator ***shall examine*** the TOE design to determine that SFR-supporting and SFR-non-
 9091 interfering modules are correctly categorised.

9092 In the cases where the developer has provided different amounts of information for different
 9093 modules, an implicit categorisation has been done. That is, modules (for instance) with detail
 9094 presented on their SFR-related interfaces (see [ADV_TDS.3.10C](#)) are candidate SFR-enforcing
 9095 modules, although examination by the evaluator may lead to a determination that some set of them
 9096 are SFR-supporting or SFR-non-interfering. Those with only a description of their purpose and
 9097 interaction with other modules (for instance) are “implicitly categorised” as SFR-supporting or
 9098 SFR-non-interfering.

In these cases, a key focus of the evaluator for this work unit is attempting to determine from the evidence provided for each module implicitly categorised as SFR-supporting or SFR-non-interfering and the evaluation information about other modules (in the TOE design, the functional specification, the security architecture description, and the operational user guidance), whether the module is indeed SFR-supporting or SFR-non-interfering. At this level of assurance some error should be tolerated; the evaluator does not have to be absolutely sure that a given module is SFR-supporting or SFR-non-interfering, even though it is labelled as such. However, if the evidence provided indicates that a SFR-supporting or SFR-non-interfering module is SFR-enforcing, the evaluator requests additional information from the developer in order to resolve the apparent inconsistency. For instance, suppose the documentation for Module A (an SFR-enforcing module) indicates that it calls Module B to perform an access check on a certain type of construct. When the evaluator examines the information associated with Module B, they find that all the developer has provided is a purpose and a set of interactions (thus implicitly categorising Module B as SFR-supporting or SFR-non-interfering). On examining the purpose and interactions from Module A, the evaluator finds no mention of Module B performing any access checks, and Module A is not listed as a module with which Module B interacts. At this point the evaluator should approach the developer to resolve the discrepancies between the information provided in Module A and that in Module B.

Another example would be where the evaluator examines the mapping of the TSFI to the modules as provided by **ADV_TDS.3.2D**. This examination shows that Module C is associated with an SFR requiring identification of the user. Again, when the evaluator examines the information associated with Module C, they find that all the developer has provided is a purpose and a set of interactions (thus implicitly categorising Module C as SFR-supporting or SFR-non-interfering). Examining the purpose and interactions presented for Module C, the evaluator is unable to determine why Module C, listed as mapping to a TSFI concerned with user identification, would not be classified as SFR-enforcing. Again, the evaluator should approach the developer to resolve this discrepancy.

A final example is from the opposite point of view. As before, the developer has provided information associated with Module D consisting of a purpose and a set of interactions (thus implicitly categorising Module D as SFR-supporting or SFR-non-interfering). The evaluator examines all of the evidence provided, including the purpose and interactions for Module D. The purpose appears to give a meaningful description of Module D's function in the TOE, the interactions are consistent with that description, and there is nothing to indicate that Module D is SFR-enforcing. In this case, the evaluator should not demand more information about Module D "just to be sure" it is correctly categorised. The developer has met their obligations and the resulting assurance the evaluator has in the implicit categorisation of Module D is (by definition) appropriate for this assurance level.

11.9.3.4.12 Work unit ADV_TDS.3-12

The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of each SFR-supporting or SFR-non-interfering module is complete and accurate.

The description of the purpose of a module indicates what function the module is fulfilling. From the description, the evaluator should be able to obtain a general idea of the module's role. In order to assure the description is complete, the evaluator uses the information provided about the module's interactions with other modules to assess whether the reasons for the module being called are consistent with the module's purpose. If the interaction description contains functionality that is not apparent from, or in conflict with, the module's purpose, the evaluator needs to determine whether the problem is one of accuracy or of completeness. The evaluator should be wary of purposes that are too short, since meaningful analysis based on a one-sentence purpose is likely to be impossible.

Because the modules are at such a low level, it may be difficult determine completeness and accuracy impacts from other documentation, such as administrative guidance, the functional specification, the security architecture description, or the TSF internals document. However, the evaluator uses the information present in those documents to the extent possible to help ensure

9151 that the function is accurately and completely described. This analysis can be aided by the analysis
 9152 performed for the work units for the ADV_TDS.3.10C element, which maps the TSFI in the
 9153 functional specification to the modules of the TSF.

9154 **11.9.3.4.13 Work unit ADV_TDS.3-13**

9155 The evaluator *shall examine* the TOE design to determine that the description of a SFR-supporting
 9156 or SFR-non-interfering module's interaction with other modules is complete and accurate.

9157 It is important to note that, in terms of the Part 3 requirement and this work unit, the term
 9158 *interaction* is intended to convey less rigour than *interface*. An interaction does not need to be
 9159 characterised at the implementation level (e.g., parameters passed from one routine in a module to
 9160 a routine in a different module; global variables; hardware signals (e.g., interrupts) from a
 9161 hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a
 9162 particular module that are going to be used by another module should be covered in this discussion.
 9163 Any control relationships between modules (e.g., a module responsible for configuring a rule base
 9164 for a firewall system and the module that actually implements these rules) should also be
 9165 described.

9166 Because the modules are at such a low level, it may be difficult determine completeness and
 9167 accuracy impacts from other documentation, such as operational user guidance, the functional
 9168 specification, the security architecture description, or the TSF internals document. However, the
 9169 evaluator uses the information present in those documents to the extent possible to help ensure
 9170 that the function is accurately and completely described. This analysis can be aided by the analysis
 9171 performed for the work units for the **ADV_TDS.3.10C** element, which maps the TSFI in the
 9172 functional specification to the modules of the TSF.

9173 A module's interaction with other modules goes beyond just a call-tree-type document. The
 9174 interaction is described from a functional perspective of why a module interacts with other
 9175 modules. The module's purpose describes what functions the module provides to other modules;
 9176 the interactions should describe what the module depends on from other modules in order to
 9177 accomplish this function.

9178 ISO/IEC 15408-3 ADV_TDS.3.10C: *The mapping shall demonstrate that all TSFIs trace to the*
 9179 *behaviour described in the TOE design that they invoke.*

9180 **11.9.3.4.14 Work unit ADV_TDS.3-14**

9181 The evaluator *shall examine* the TOE design to determine that it contains a complete and accurate
 9182 mapping from the TSFI described in the functional specification to the modules of the TSF
 9183 described in the TOE design.

9184 The modules described in the TOE design provide a description of the implementation of the TSF.
 9185 The TSFI provide a description of how the implementation is exercised. The evidence from the
 9186 developer identifies the module that is initially invoked when an operation is requested at the TSFI,
 9187 and identifies the chain of modules invoked up to the module that is primarily responsible for
 9188 implementing the functionality. However, a complete call tree for each TSFI is not required for this
 9189 work unit. The cases in which more than one module would have to be identified are where there
 9190 are "entry point" modules or wrapper modules that have no functionality other than conditioning
 9191 inputs or de-multiplexing an input. Mapping to one of these modules would not provide any useful
 9192 information to the evaluator.

9193 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at
 9194 least one module. The verification of accuracy is more complex.

9195 The first aspect of accuracy is that each TSFI is mapped to a module at the TSF boundary. This
 9196 determination can be made by reviewing the module description and its interfaces/interactions.
 9197 The next aspect of accuracy is that each TSFI identifies a chain of modules between the initial

9198 module identified and a module that is primarily responsible for implementing the function
 9199 presented at the TSF. Note that this may be the initial module, or there may be several modules,
 9200 depending on how much pre-conditioning of the inputs is done. It should be noted that one
 9201 indicator of a pre-conditioning module is that it is invoked for a large number of the TSFI, where
 9202 the TSFI are all of similar type (e.g., system call). The final aspect of accuracy is that the mapping
 9203 makes sense. For instance, mapping a TSFI dealing with access control to a module that checks
 9204 passwords is not accurate. The evaluator should again use judgement in making this determination.
 9205 The goal is that this information aids the evaluator in understanding the system and
 9206 implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the
 9207 TSF. The bulk of the assessment of whether the SFRs are described accurately by the modules is
 9208 performed in other work units.

9209 **11.9.3.5 Action ADV_TDS.3.2E**

9210 **11.9.3.5.1 Work unit ADV_TDS.3-15**

9211 The evaluator *shall examine* the TOE security functional requirements and the TOE design, to
 9212 determine that all ST security functional requirements are covered by the TOE design.

9213 The evaluator may construct a map between the TOE security functional requirements and the TOE
 9214 design. This map will likely be from a functional requirement to a set of subsystems, and later to
 9215 modules. Note that this map may have to be at a level of detail below the component or even
 9216 element level of the requirements, because of operations (assignments, refinements, selections)
 9217 performed on the functional requirement by the ST author.

9218 For example, the **FDP_ACC.1 Subset access control** component contains an element with
 9219 assignments. If the ST contained, for instance, ten rules in the **FDP_ACC.1 Subset access control**
 9220 assignment, and these ten rules were implemented in specific places within fifteen modules, it
 9221 would be inadequate for the evaluator to map **FDP_ACC.1 Subset access control** to one subsystem
 9222 and claim the work unit had been completed. Instead, the evaluator would map **FDP_ACC.1 Subset**
 9223 **access control** (rule 1) to modules x, y, and z of subsystem A; **FDP_ACC.1 Subset access control** (rule
 9224 2) to modules x, p, and q of subsystem A; etc.

9225 **11.9.3.5.2 Work unit ADV_TDS.3-16**

9226 The evaluator *shall examine* the TOE design to determine that it is an accurate instantiation of all
 9227 security functional requirements.

9228 The evaluator may construct a map between the TOE security functional requirements and the TOE
 9229 design. This map will likely be from a functional requirement to a set of subsystems. Note that this
 9230 map may have to be at a level of detail below the component or even element level of the
 9231 requirements, because of operations (assignments, refinements, selections) performed on the
 9232 functional requirement by the ST author.

9233 As an example, if the ST requirements specified a role-based access control mechanism, the
 9234 evaluator would first identify the subsystems, and modules that contribute to this mechanism's
 9235 implementation. This could be done by in-depth knowledge or understanding of the TOE design or
 9236 by work done in the previous work unit. Note that this trace is only to identify the subsystems, and
 9237 modules, and is not the complete analysis.

9238 The next step would be to understand what mechanism the subsystems, and modules implemented.
 9239 For instance, if the design described an implementation of access control based on UNIX-style
 9240 protection bits, the design would not be an accurate instantiation of those access control
 9241 requirements present in the ST example used above. If the evaluator could not determine that the
 9242 mechanism was accurately implemented because of a lack of detail, the evaluator would have to
 9243 assess whether all of the SFR-enforcing subsystems and modules have been identified, or if
 9244 adequate detail had been provided for those subsystems and modules.

9245 **11.9.4 Evaluation of sub-activity (ADV_TDS.4)**

9246 **11.9.4.1 Objectives**

9247 The objective of this sub-activity is to determine whether the TOE design provides a description of
 9248 the TOE in terms of subsystems sufficient to determine the TSF boundary, and provides a
 9249 description of the TSF internals in terms of modules (and optionally higher-level abstractions). It
 9250 provides a detailed description of the SFR-enforcing and SFR-supporting modules and enough
 9251 information about the SFR-non-interfering modules for the evaluator to determine that the SFRs
 9252 are completely and accurately implemented; as such, the TOE design provides an explanation of the
 9253 implementation representation.

9254 **11.9.4.2 Input**

9255 The evaluation evidence for this sub-activity is:

- 9256 a) the ST;
- 9257 b) the functional specification;
- 9258 c) security architecture description;
- 9259 d) the TOE design.

9260 **11.9.4.3 Application notes**

9261 There are three types of activity that the evaluator must undertake with respect to the TOE design.
 9262 First, the evaluator determines that the TSF boundary has been adequately described. Second, the
 9263 evaluator determines that the developer has provided documentation that conforms to the content
 9264 and presentation requirements this subsystem, and that is consistent with other documentation
 9265 provided for the TOE. Finally, the evaluator must analyse the design information provided for the
 9266 SFR-enforcing modules (at a detailed level) and the SFR-supporting and SFR-non-interfering
 9267 modules (at a less detailed level) to understand how the system is implemented, and with that
 9268 knowledge ensure that the TSFI in the functional specification are adequately described, and that
 9269 the test information adequately tests the TSF (done in the Class ATE: Tests work units).

9270 **11.9.4.4 Action ADV_TDS.4.1E**

9271 ISO/IEC 15408-3 ADV_TDS.4.1C: *The design shall describe the structure of the TOE in terms of*
 9272 *subsystems.*

9273 **11.9.4.4.1 Work unit ADV_TDS.4-1**

9274 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is
 9275 described in terms of subsystems.

9276 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the
 9277 TOE will be used as input to work unit ADV_TDS.4-4, where the parts of the TOE that make up the
 9278 TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

9279 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and
 9280 modules) Depending upon the complexity of the TOE, its design may be described in terms of
 9281 subsystems and modules, as described in ISO/IEC 15408-3 **Annex A.4, ADV_TDS: Subsystems and**
 9282 **Modules**. For a very simple TOE that can be described solely at the “module” level (see ADV_TDS.4-
 9283 2), this work unit is not applicable and therefore considered to be satisfied.

9284 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST,
9285 operator user guidance) to determine that the description of the TOE in such evidence is consistent
9286 with the description contained in the TOE design.

9287 ISO/IEC 15408-3 ADV_TDS.4.2C: *The design shall describe the TSF in terms of modules, designating*
9288 *each module as SFR-enforcing, SFR-supporting, or SFR-non-interfering.*

9289 11.9.4.4.2 Work unit ADV_TDS.4-2

9290 The evaluator **shall examine** the TOE design to determine that the entire TSF is described in terms
9291 of modules.

9292 The evaluator will examine the modules for specific properties in other work units; in this work
9293 unit the evaluator determines that the modular description covers the entire TSF, and not just a
9294 portion of the TSF. The evaluator uses other evidence provided for the evaluation (e.g., functional
9295 specification, architectural description) in making this determination. For example, if the functional
9296 specification contains interfaces to functionality that does not appear to be described in the TOE
9297 design description, it may be the case that a portion of the TSF has not been included appropriately.
9298 Making this determination will likely be an iterative process, where as more analysis is done on the
9299 other evidence, more confidence can be gained with respect to the completeness of the
9300 documentation.

9301 Unlike subsystems, modules describe the implementation in a level of detail that can serve as a
9302 guide to reviewing the implementation representation. A description of a module should be such
9303 that one could create an implementation of the module from the description, and the resulting
9304 implementation would be 1) identical to the actual TSF implementation in terms of the interfaces
9305 presented, 2) identical in the use of interfaces that are mentioned in the design, and 3) functionally
9306 equivalent to the description of the purpose of the TSF module. For instance, RFC 793 provides a
9307 high-level description of the TCP protocol. It is necessarily implementation independent. While it
9308 provides a wealth of detail, it is **not** a suitable design description because it is not specific to an
9309 implementation. An actual implementation can add to the protocol specified in the RFC, and
9310 implementation choices (for instance, the use of global data vs. local data in various parts of the
9311 implementation) may have an impact on the analysis that is performed. The design description of
9312 the TCP module would list the interfaces presented by the implementation (rather than just those
9313 defined in RFC 793), as well as an algorithm description of the processing associated with the
9314 modules implementing TCP (assuming it was part of the TSF).

9315 11.9.4.4.3 Work unit ADV_TDS.4-3

9316 The evaluator **shall check** the TOE design to determine that the TSF modules are identified as
9317 either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

9318 The purpose of designating each module (according to the role a particular module plays in the
9319 enforcement of the SFRs) is to allow developers to provide less information about the parts of the
9320 TSF that have little role in security. It is always permissible for the developer to provide more
9321 information or detail than the requirements demand, as might occur when the information has
9322 been gathered outside the evaluation context. In such cases the developer must still designate the
9323 modules as either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

9324 The accuracy of these designations is continuously reviewed as the evaluation progresses. The
9325 concern is the mis-designation of modules as being less important (and hence, having less
9326 information) than is really the case. While blatant mis-designations may be immediately apparent
9327 (e.g., designating an authentication module as anything but SFR-enforcing when **User identification**
9328 **(FIA_UID)** is one of the SFRs being claimed), other mis-designations might not be discovered until
9329 the TSF is better understood. The evaluator must therefore keep in mind that these designations
9330 are the developer's initial best effort, but are subject to change. Further guidance is provided under
9331 work unit ADV_TDS.4-17, which examines the accuracy of these designations.

- 9332 ISO/IEC 15408-3 ADV_TDS.4.3C: *The design shall identify all subsystems of the TSF.*
- 9333 **11.9.4.4.4 Work unit ADV_TDS.4-4**
- 9334 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are
9335 identified.
- 9336 If the design is presented solely in terms of modules, then subsystems in these requirements are
9337 equivalent to modules and the activity should be performed at the module level.
- 9338 In work unit ADV_TDS.4-1 all of the subsystems of the TOE were identified, and a determination
9339 made that the non-TSF subsystems were correctly characterised. Building on that work, the
9340 subsystems that were not characterised as non-TSF subsystems should be precisely identified. The
9341 evaluator determines that, of the hardware and software installed and configured according to the
9342 Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one
9343 that is part of the TSF, or one that is not.
- 9344 ISO/IEC 15408-3 ADV_TDS.4.4C: *The design shall provide a semiformal description of each*
9345 *subsystem of the TSF, supported by informal, explanatory text where appropriate.*
- 9346 **11.9.4.4.5 Work unit ADV_TDS.4-5**
- 9347 The evaluator ***shall examine*** the TDS documentation to determine that the semiformal notation
9348 used for describing the subsystems, modules and their interfaces is defined or referenced.
- 9349 A semiformal notation can be either defined by the sponsor or a corresponding standard be
9350 referenced. The evaluator should provide a mapping of security functions and their interfaces
9351 outlining in what part of the documentation a function or interface is semiformal described and
9352 what notation is used. The evaluator examines all semiformal notations used to make sure that
9353 they are of a semiformal style and to justify the appropriateness of the manner how the semiformal
9354 notations are used for the TOE.
- 9355 The evaluator is reminded that a semi-formal presentation is characterised by a standardised
9356 format with a well-defined syntax that reduces ambiguity that may occur in informal presentations.
9357 The syntax of all semiformal notations used in the functional specification shall be defined or a
9358 corresponding standard be referenced. The evaluator verifies that the semiformal notations used
9359 for expressing the functional specification are capable of expressing features relevant to security.
9360 In order to determine this, the evaluator can refer to the SFR and compare the TSF security
9361 features stated in the ST and those described in the FSP using the semiformal notations.
- 9362 **11.9.4.4.6 Work unit ADV_TDS.4-6**
- 9363 The evaluator ***shall examine*** the TOE design to determine that each subsystem of the TSF
9364 describes its role in the enforcement of SFRs described in the ST.
- 9365 If the design is presented solely in terms of modules, then this work unit will be considered
9366 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
9367 evaluator is necessary in this case.
- 9368 On systems that are complex enough to warrant a subsystem-level description of the TSF in
9369 addition to the modular description, the goal of the subsystem-level description is to give the
9370 evaluator context for the modular description that follows. Therefore, the evaluator ensures that
9371 the subsystem-level description contains a description of how the security functional requirements
9372 are achieved in the design, but at a level of abstraction above the modular description. This
9373 description should discuss the mechanisms used at a level that is aligned with the module
9374 description; this will provide the evaluators the road map needed to intelligently assess the
9375 information contained in the module description. A well-written set of subsystem descriptions will
9376 help guide the evaluator in determining the modules that are most important to examine, thus

9377 focusing the evaluation activity on the portions of the TSF that have the most relevance with
9378 respect to the enforcement of the SFRs.

9379 The evaluator ensures that all subsystems of the TSF have a description. While the description
9380 should focus on the role that the subsystem plays in enforcing or supporting the implementation of
9381 the SFRs, enough information must be present so that a context for understanding the SFR-related
9382 functionality is provided.

9383 **11.9.4.4.7 Work unit ADV_TDS.4-7**

9384 The evaluator ***shall examine*** the TOE design to determine that each SFR-non-interfering
9385 subsystem of the TSF is described such that the evaluator can determine that the subsystem is SFR-
9386 non-interfering.

9387 If the design is presented solely in terms of modules, then this work unit will be considered
9388 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
9389 evaluator is necessary in this case.

9390 An SFR-non-interfering subsystem is one on which the SFR-enforcing and SFR-supporting
9391 subsystems have no dependence; that is, they play no role in implementing SFR functionality.

9392 The evaluator ensures that all subsystems of the TSF have a description. While the description
9393 should focus on the role that the subsystem do not plays in enforcing or supporting the
9394 implementation of the SFRs, enough information must be present so that a context for
9395 understanding the SFR-non-interfering functionality is provided.

9396 ISO/IEC 15408-3 ADV_TDS.4.5C: *The design shall provide a description of the interactions among all*
9397 *subsystems of the TSF.*

9398 **11.9.4.4.8 Work unit ADV_TDS.4-8**

9399 The evaluator ***shall examine*** the TOE design to determine that interactions between the
9400 subsystems of the TSF are described.

9401 If the design is presented solely in terms of modules, then this work unit will be considered
9402 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
9403 evaluator is necessary in this case.

9404 On systems that are complex enough to warrant a subsystem-level description of the TSF in
9405 addition to the modular description, the goal of describing the interactions between the
9406 subsystems is to help provide the reader a better understanding of how the TSF performs its
9407 functions. These interactions do not need to be characterised at the implementation level (e.g.,
9408 parameters passed from one routine in a subsystem to a routine in a different subsystem; global
9409 variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling
9410 subsystem), but the data elements identified for a particular subsystem that are going to be used by
9411 another subsystem need to be covered in this discussion. Any control relationships between
9412 subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the
9413 subsystem that actually implements these rules) should also be described.

9414 It should be noted while the developer should characterise all interactions between subsystems,
9415 the evaluators need to use their own judgement in assessing the completeness of the description. If
9416 the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for
9417 instance, in examining the module-level documentation) that do not appear to be described, the
9418 evaluator ensures that this information is provided by the developer. However, if the evaluator can
9419 determine that interactions among a particular set of subsystems, while incompletely described by
9420 the developer, and a complete description will not aid in understanding the overall functionality
9421 nor security functionality provided by the TSF, then the evaluator may choose to consider the
9422 description sufficient, and not pursue completeness for its own sake.

9423 ISO/IEC 15408-3 ADV_TDS.4.6C: *The design shall provide a mapping from the subsystems of the TSF*
 9424 *to the modules of the TSF.*

9425 **11.9.4.4.9 Work unit ADV_TDS.4-9**

9426 The evaluator ***shall examine*** the TOE design to determine that the mapping between the
 9427 subsystems of the TSF and the modules of the TSF is complete.

9428 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

9429 For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition
 9430 to the modular description, the developer provides a simple mapping showing how the modules of
 9431 the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their
 9432 module-level assessment. To determine completeness, the evaluator examines each mapping and
 9433 determines that all subsystems map to at least one module, and that all modules map to exactly one
 9434 subsystem.

9435 **11.9.4.4.10 Work unit ADV_TDS.4-10**

9436 The evaluator ***shall examine*** the TOE design to determine that the mapping between the
 9437 subsystems of the TSF to the modules of the TSF is accurate.

9438 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

9439 For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition
 9440 to the modular description, the developer provides a simple mapping showing how the modules of
 9441 the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their
 9442 module-level assessment. The evaluator may choose to check the accuracy of the mapping in
 9443 conjunction with performing other work units. An “inaccurate” mapping is one where the module
 9444 is mistakenly associated with a subsystem where its functions are not used within the subsystem.
 9445 Because the mapping is intended to be a guide supporting more detailed analysis, the evaluator is
 9446 cautioned to apply appropriate effort to this work unit. Expending extensive evaluator resources
 9447 verifying the accuracy of the mapping is not necessary. Inaccuracies that lead to mis-
 9448 understandings related to the design that are uncovered as part of this or other work units are the
 9449 ones that should be associated with this work unit and corrected.

9450 ISO/IEC 15408-3 ADV_TDS.4.7C: *The design shall describe each SFR-enforcing and SFR-supporting*
 9451 *module in terms of its purpose and relationship with other modules.*

9452 **11.9.4.4.11 Work unit ADV_TDS.4-11**

9453 The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of
 9454 each SFR-enforcing and SFR-supporting module, and relationship with other modules is complete
 9455 and accurate.

9456 The developer may designate modules as SFR-enforcing, SFR-supporting, and SFR non-interfering,
 9457 but these “tags” are used only to describe the amount and type of information the developer must
 9458 provide, and can be used to limit the amount of information the developer has to develop if their
 9459 engineering process does not produce the documentation required. Whether the modules have
 9460 been categorised by the developer or not, it is the evaluator's responsibility to determine that the
 9461 modules have the appropriate information for their role (SFR-enforcing, etc.) in the TOE, and to
 9462 obtain the appropriate information from the developer should the developer fail to provide the
 9463 required information for a particular module.

9464 The purpose of a module provides a description indicating what function the module is fulfilling. A
 9465 word of caution to evaluator is in order. The focus of this work unit should be to provide the
 9466 evaluator an understanding of how the module works so that determinations can be made about
 9467 the soundness of the implementation of the SFRs, as well as to support architectural analysis

9468 performed for ADV_ARC subsystems. As long as the evaluator has a sound understanding of the
 9469 module's operation, and its relationship to other modules and the TOE as a whole, the evaluator
 9470 should consider the objective of the work achieved and not engage in a documentation exercise for
 9471 the developer (by requiring, for example, a complete algorithmic description for a self-evident
 9472 implementation representation).

9473 Because the modules are at such a low level, it may be difficult determine completeness and
 9474 accuracy impacts from other documentation, such as operational user guidance, the functional
 9475 specification, the TSF internals, or the security architecture description. However, the evaluator
 9476 uses the information present in those documents to the extent possible to help ensure that the
 9477 purpose is accurately and completely described. This analysis can be aided by the analysis
 9478 performed for the work units for the **ADV_TDS.4.10C** element, which maps the TSFI in the
 9479 functional specification to the modules of the TSF.

9480 ISO/IEC 15408-3 ADV_TDS.4.8C: *The design shall describe each SFR-enforcing and SFR-supporting*
 9481 *module in terms of its SFR-related interfaces, return values from those interfaces, interaction with*
 9482 *other modules and called SFR-related interfaces to other SFR-enforcing or SFR-supporting modules.*

9483 **11.9.4.4.12 Work unit ADV_TDS.4-12**

9484 The evaluator ***shall examine*** the TOE design to determine that the description of the interfaces
 9485 presented by each SFR-enforcing and SFR-supporting module contain an accurate and complete
 9486 description of the SFR-related parameters, the invocation conventions for each interface, and any
 9487 values returned directly by the interface.

9488 The SFR-related interfaces of a module are those interfaces used by other modules as a means to
 9489 invoke the SFR-related operations provided, and to provide inputs to or receive outputs from the
 9490 module. The purpose in the specification of these interfaces is to permit the exercise of them
 9491 during testing. Inter-module interfaces that are not SFR-related need not be specified or described,
 9492 since they are not a factor in testing. Likewise, other internal interfaces that are not a factor in
 9493 traversing SFR-related paths of execution (such as those internal paths that are fixed).

9494 SFR-related interfaces of SFR-supporting modules are all interfaces of SFR-supporting modules
 9495 that are called directly or indirectly from SFR-enforcing modules. Those interfaces need to be
 9496 described with all the parameter used in such a call. This allows the evaluator to understand the
 9497 purpose of the call to the SFR-supporting module in the context of operation of the SFR-enforcing
 9498 modules.

9499 SFR-related interfaces are described in terms of how they are invoked, and any values that are
 9500 returned. This description would include a list of parameters, and descriptions of these parameters.
 9501 Note that global data would also be considered parameters if used by the module (either as inputs
 9502 or outputs) when invoked. If a parameter were expected to take on a set of values (e.g., a “flag”
 9503 parameter), the complete set of values the parameter could take on that would have an effect on
 9504 module processing would be specified. Likewise, parameters representing data structures are
 9505 described such that each field of the data structure is identified and described. Note that different
 9506 programming languages may have additional “interfaces” that would be non-obvious; an example
 9507 would be operator/function overloading in C++. This “implicit interface” in the class description
 9508 would also be described as part of the low-level TOE design. Note that although a module could
 9509 present only one interface, it is more common that a module presents a small set of related
 9510 interfaces.

9511 In terms of the assessment of parameters (inputs and outputs) to a module, any use of global data
 9512 must also be considered. A module “uses” global data if it either reads or writes the data. In order
 9513 to assure the description of such parameters (if used) is complete, the evaluator uses other
 9514 information provided about the module in the TOE design (interfaces, algorithmic description, etc.),
 9515 as well as the description of the particular set of global data assessed in work unit ADV_TDS.4-12.
 9516 For instance, the evaluator could first determine the processing the module performs by examining
 9517 its function and interfaces presented (particularly the parameters of the interfaces). They could

9518 then check to see if the processing appears to “touch” any of the global data areas identified in the
 9519 TDS design. The evaluator then determines that, for each global data area that appears to be
 9520 “touched”, that global data area is listed as a means of input or output by the module the evaluator
 9521 is examining.

9522 Invocation conventions are a programming-reference-type description that one could use to
 9523 correctly invoke a module's interface if one were writing a program to make use of the module's
 9524 functionality through that interface. This includes necessary inputs and outputs, including any set-
 9525 up that may need to be performed with respect to global variables.

9526 Values returned through the interface refer to values that are either passed through parameters or
 9527 messages; values that the function call itself returns in the style of a “C” program function call; or
 9528 values passed through global means (such as certain error routines in *ix-style operating systems).

9529 In order to assure the description is complete, the evaluator uses other information provided about
 9530 the module in the TOE design (e.g., algorithmic description, global data used) to ensure that it
 9531 appears all data necessary for performing the functions of the module is presented to the module,
 9532 and that any values that other modules expect the module under examination to provide are
 9533 identified as being returned by the module. The evaluator determines accuracy by ensuring that
 9534 the description of the processing matches the information listed as being passed to or from an
 9535 interface.

9536 ISO/IEC 15408-3 ADV_TDS.4.9C: *The design shall describe each SFR-non-interfering module in terms*
 9537 *of its purpose and interaction with other modules.*

9538 **11.9.4.4.13 Work unit ADV_TDS.4-13**

9539 The evaluator ***shall examine*** the TOE design to determine that SFR-non-interfering modules are
 9540 correctly categorised.

9541 As mentioned in work unit ADV_TDS.4-2, less information is required about modules that are SFR-
 9542 non-interfering. A key focus of the evaluator for this work unit is attempting to determine from the
 9543 evidence provided for each module implicitly categorised as SFR-non-interfering and the
 9544 evaluation (information about other modules in the TOE design, the functional specification, the
 9545 security architecture description, the operational user guidance, the TSF internals document, and
 9546 perhaps even the implementation representation) whether the module is indeed SFR-non-
 9547 interfering. At this level of assurance some error should be tolerated; the evaluator does not have
 9548 to be absolutely sure that a given module is SFR-non-interfering, even though it is labelled as such.
 9549 However, if the evidence provided indicates that a SFR-non-interfering module is SFR-enforcing or
 9550 SFR-supporting, the evaluator requests additional information from the developer in order to
 9551 resolve the apparent inconsistency. For example, suppose the documentation for Module A (an
 9552 SFR-enforcing module) indicates that it calls Module B to perform an access check on a certain type
 9553 of construct. When the evaluator examines the information associated with Module B, it is
 9554 discovered that the only information the developer has provided is a purpose and a set of
 9555 interactions (thus implicitly categorising Module B as SFR-supporting or SFR-non-interfering). On
 9556 examining the purpose and interactions from Module A, the evaluator finds no mention of Module
 9557 B performing any access checks, and Module A is not listed as a module with which Module B
 9558 interacts. At this point the evaluator should approach the developer to resolve the discrepancies
 9559 between the information provided in Module A and that in Module B.

9560 Another example would be where the evaluator examines the mapping of the TSFI to the modules
 9561 as provided by **ADV_TDS.4.2D**. This examination shows that Module C is associated with an SFR
 9562 requiring identification of the user. Again, when the evaluator examines the information associated
 9563 with Module C, they find that all the developer has provided is a purpose and a set of interactions
 9564 (thus implicitly categorising Module C as SFR-non-interfering). Examining the purpose and
 9565 interactions presented for Module C, the evaluator is unable to determine why Module C, listed as
 9566 mapping to a TSFI concerned with user identification, would not be classified as SFR-enforcing or
 9567 SFR-supporting. Again, the evaluator should approach the developer to resolve this discrepancy.

A final example illustrates the opposite situation. As before, the developer has provided information associated with Module D consisting of a purpose and a set of interactions (thus implicitly categorising Module D as SFR-non-interfering). The evaluator examines all of the evidence provided, including the purpose and interactions for Module D. The purpose appears to give a meaningful description of Module D's function in the TOE, the interactions are consistent with that description, and there is nothing to indicate that Module D is SFR-enforcing or SFR-supporting. In this case, the evaluator should not demand more information about Module D "just be to sure" it is correctly categorised. The developer has met the obligations and the resulting assurance the evaluator has in the implicit categorisation of Module D is (by definition) appropriate for this assurance level.

11.9.4.4.14 Work unit ADV_TDS.4-14

The evaluator *shall examine* the TOE design to determine that the description of the purpose of each SFR-non-interfering module is complete and accurate.

The description of the purpose of a module indicates what function the module is fulfilling. From the description, the evaluator should be able to obtain a general idea of the module's role. In order to assure the description is complete, the evaluator uses the information provided about the module's interactions with other modules to assess whether the reasons for the module being called are consistent with the module's purpose. If the interaction description contains functionality that is not apparent from, or in conflict with, the module's purpose, the evaluator needs to determine whether the problem is one of accuracy or of completeness. The evaluator should be wary of purposes that are too short, since meaningful analysis based on a one-sentence purpose is likely to be impossible.

Because the modules are at such a low level, it may be difficult determine completeness and accuracy impacts from other documentation, such as operational user guidance, the functional specification, the security architecture description, or the TSF internals document. However, the evaluator uses the information present in those documents to the extent possible to help ensure that the function is accurately and completely described. This analysis can be aided by the analysis performed for the work units for the **ADV_TDS.4.10C** element, which maps the TSFI in the functional specification to the modules of the TSF.

11.9.4.4.15 Work unit ADV_TDS.4-15

The evaluator *shall examine* the TOE design to determine that the description of a SFR-non-interfering module's interaction with other modules is complete and accurate.

It is important to note that, in terms of the Part 3 requirement and this work unit, the term *interaction* is intended to convey less rigour than *interface*. An interaction does not need to be characterised at the implementation level (e.g., parameters passed from one routine in a module to a routine in a different module; global variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a particular module that are going to be used by another module should be covered in this discussion. Any control relationships between modules (e.g., a module responsible for configuring a rule base for a firewall system and the module that actually implements these rules) should also be described.

A module's interaction with other modules can be captured in many ways. The intent for the TOE design is to allow the evaluator to understand (in part through analysis of module interactions) the role of the SFR-supporting and SFR-non-interfering modules in the overall TOE design. Understanding of this role will aid the evaluator in performing work unit ADV_TDS.4-8.

A module's interaction with other modules goes beyond just a call-tree-type document. The interaction is described from a functional perspective of why a module interacts with other modules. The module's purpose describes what functions the module provides to other modules;

9616 the interactions should describe what the module depends on from other modules in order to
9617 accomplish this function.

9618 Because the modules are at such a low level, it may be difficult to determine completeness and
9619 accuracy impacts from other documentation, such as operational user guidance, the functional
9620 specification, the security architecture description, or the TSF internals document. However, the
9621 evaluator uses the information present in those documents to the extent possible to help ensure
9622 that the interactions are accurately and completely described.

9623 ISO/IEC 15408-3 ADV_TDS.4.10C: *The mapping shall demonstrate that all TSFIs trace to the*
9624 *behaviour described in the TOE design that they invoke.*

9625 **11.9.4.4.16 Work unit ADV_TDS.4-16**

9626 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate
9627 mapping from the TSFI described in the functional specification to the modules of the TSF
9628 described in the TOE design.

9629 The modules described in the TOE design provide a description of the implementation of the TSF.
9630 The TSFI provide a description of how the implementation is exercised. The evidence from the
9631 developer identifies the module that is initially invoked when an operation is requested at the TSFI,
9632 and identify the chain of modules invoked up to the module that is primarily responsible for
9633 implementing the functionality. However, a complete call tree for each TSFI is not required for this
9634 work unit. The cases in which more than one module would have to be identified are where there
9635 are “entry point” modules or wrapper modules that have no functionality other than conditioning
9636 inputs or de-multiplexing an input. Mapping to one of these modules would not provide any useful
9637 information to the evaluator.

9638 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at
9639 least one module. The verification of accuracy is more complex.

9640 The first aspect of accuracy is that each TSFI is mapped to a module at the TSF boundary. This
9641 determination can be made by reviewing the module description and its interfaces/interactions.
9642 The next aspect of accuracy is that each TSFI identifies a chain of modules between the initial
9643 module identified and a module that is primarily responsible for implementing the function
9644 presented at the TSF. Note that this may be the initial module, or there may be several modules,
9645 depending on how much pre-conditioning of the inputs is done. It should be noted that one
9646 indicator of a pre-conditioning module is that it is invoked for a large number of the TSFI, where
9647 the TSFI are all of similar type (e.g., system call). The final aspect of accuracy is that the mapping
9648 makes sense. For instance, mapping a TSFI dealing with access control to a module that checks
9649 passwords is not accurate. The evaluator should again use judgement in making this determination.
9650 The goal is that this information aids the evaluator in understanding the system and
9651 implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the
9652 TSF. The bulk of the assessment of whether the SFRs are described accurately by the modules is
9653 performed in other work units.

9654 **11.9.4.5 Action ADV_TDS.4.2E**

9655 **11.9.4.5.1 Work unit ADV_TDS.4-17**

9656 The evaluator ***shall examine*** the TOE security functional requirements and the TOE design, to
9657 determine that all ST security functional requirements are covered by the TOE design.

9658 The evaluator may construct a map between the TOE security functional requirements and the TOE
9659 design. This map will likely be from a functional requirement to a set of subsystems, and later to
9660 modules. Note that this map may have to be at a level of detail below the component or even
9661 element level of the requirements, because of operations (assignments, refinements, selections)
9662 performed on the functional requirement by the ST author.

9663 For example, the **FDP_ACC.1 Subset access control** component contains an element with
 9664 assignments. If the ST contained, for instance, ten rules in the **FDP_ACC.1 Subset access control**
 9665 assignment, and these ten rules were implemented in specific places within fifteen modules, it
 9666 would be inadequate for the evaluator to map **FDP_ACC.1 Subset access control** to one subsystem
 9667 and claim the work unit had been completed. Instead, the evaluator would map **FDP_ACC.1 Subset**
 9668 **access control** (rule 1) to modules x, y and z of subsystem A; **FDP_ACC.1 Subset access control** (rule
 9669 2) to x, p, and q of subsystem A; etc.

9670 **11.9.4.5.2 Work unit ADV_TDS.4-18**

9671 The evaluator ***shall examine*** the TOE design to determine that it is an accurate instantiation of all
 9672 security functional requirements.

9673 The evaluator may construct a map between the TOE security functional requirements and the TOE
 9674 design. This map will likely be from a functional requirement to a set of subsystems. Note that this
 9675 map may have to be at a level of detail below the component or even element level of the
 9676 requirements, because of operations (assignments, refinements, selections) performed on the
 9677 functional requirement by the ST author.

9678 As an example, if the ST requirements specified a role-based access control mechanism, the
 9679 evaluator would first identify the subsystems, and modules that contribute to this mechanism's
 9680 implementation. This could be done by in-depth knowledge or understanding of the TOE design or
 9681 by work done in the previous work unit. Note that this trace is only to identify the subsystems, and
 9682 modules, and is not the complete analysis.

9683 The next step would be to understand what mechanism the subsystems, and modules implemented.
 9684 For instance, if the design described an implementation of access control based on UNIX-style
 9685 protection bits, the design would not be an accurate instantiation of those access control
 9686 requirements present in the ST example used above. If the evaluator could not determine that the
 9687 mechanism was accurately implemented because of a lack of detail, the evaluator would have to
 9688 assess whether all of the SFR-enforcing subsystems and modules have been identified, or if
 9689 adequate detail had been provided for those subsystems and modules.

9690 **11.9.5 Evaluation of sub-activity (ADV_TDS.5)**

9691 **11.9.5.1 Objectives**

9692 The objectives of this sub-activity are to determine whether the TOE design provides a description
 9693 of the TOE in terms of subsystems sufficient to determine the TSF boundary, and provides a
 9694 description of the TSF internals in terms of modules (and optionally higher-level abstractions). It
 9695 provides enough information about the modules for the evaluator to determine that the SFRs are
 9696 completely and accurately implemented; as such, the TOE design provides an explanation of the
 9697 implementation representation.

9698 **11.9.5.2 Input**

9699 The evaluation evidence for this sub-activity is:

- 9700 a) the ST;
- 9701 b) the functional specification;
- 9702 c) security architecture description;
- 9703 d) the TOE design.

9704 **11.9.5.3 Application notes**

9705 There are three types of activity that the evaluator must undertake with respect to the TOE design.
 9706 First, the evaluator determines that the TSF boundary has been adequately described. Second, the
 9707 evaluator determines that the developer has provided documentation that conforms to the content
 9708 and presentation requirements this subsystem, and that is consistent with other documentation
 9709 provided for the TOE. Finally, the evaluator must analyse the design information provided for the
 9710 modules (at a detailed level) to understand how the system is implemented, and with that
 9711 knowledge ensure that the TSFI in the functional specification are adequately described, and that
 9712 the test information adequately tests the TSF (done in the Class ATE: Tests work units).

9713 **11.9.5.4 Action ADV_TDS.5.1E**

9714 **ADV_TDS.5.1C** *The design shall describe the structure of the TOE in terms of*
 9715 *subsystems.*

9716 **11.9.5.4.1 Work unit ADV_TDS.5-1**

9717 The evaluator **shall examine** the TOE design to determine that the structure of the entire TOE is
 9718 described in terms of subsystems.

9719 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the
 9720 TOE will be used as input to work unit ADV_TDS.5-4, where the parts of the TOE that make up the
 9721 TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

9722 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and
 9723 modules) Depending upon the complexity of the TOE, its design may be described in terms of
 9724 subsystems and modules, as described in CC Part 3 Annex A.4, ADV_TDS: Subsystems and Modules.
 9725 For a very simple TOE that can be described solely at the "module" level (see ADV_TDS.5-2), this
 9726 work unit is not applicable and therefore considered to be satisfied.

9727 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST,
 9728 operator user guidance) to determine that the description of the TOE in such evidence is consistent
 9729 with the description contained in the TOE design.

9730 **ADV_TDS.5.2C** *The design shall describe the TSF in terms of modules, designating each*
 9731 *module as SFR-enforcing, SFR-supporting, or SFR-non-interfering.*

9732 **11.9.5.4.2 Work unit ADV_TDS.5-2**

9733 The evaluator **shall examine** the TOE design to determine that the entire TSF is described in terms
 9734 of modules.

9735 The evaluator will examine the modules for specific properties in other work units; in this work
 9736 unit the evaluator determines that the modular description covers the entire TSF, and not just a
 9737 portion of the TSF. The evaluator uses other evidence provided for the evaluation (e.g., functional
 9738 specification, architectural description) in making this determination. For example, if the functional
 9739 specification contains interfaces to functionality that does not appear to be described in the TOE
 9740 design description, it may be the case that a portion of the TSF has not been included appropriately.
 9741 Making this determination will likely be an iterative process, where as more analysis is done on the
 9742 other evidence, more confidence can be gained with respect to the completeness of the
 9743 documentation.

9744 Unlike subsystems, modules describe the implementation in a level of detail that can serve as a
 9745 guide to reviewing the implementation representation. A description of a module should be such
 9746 that one could create an implementation of the module from the description, and the resulting
 9747 implementation would be 1) identical to the actual TSF implementation in terms of the interfaces
 9748 presented, 2) identical in the use of interfaces that are mentioned in the design, and 3) functionally

9749 equivalent to the description of the purpose of the TSF module. For instance, RFC 793 provides a
 9750 high-level description of the TCP protocol. It is necessarily implementation independent. While it
 9751 provides a wealth of detail, it is **not** a suitable design description because it is not specific to an
 9752 implementation. An actual implementation can add to the protocol specified in the RFC, and
 9753 implementation choices (for instance, the use of global data vs. local data in various parts of the
 9754 implementation) may have an impact on the analysis that is performed. The design description of
 9755 the TCP module would list the interfaces presented by the implementation (rather than just those
 9756 defined in RFC 793), as well as an algorithm description of the processing associated with the
 9757 modules implementing TCP (assuming it was part of the TSF).

9758 **11.9.5.4.3 Work unit ADV_TDS.5-3**

9759 The evaluator **shall check** the TOE design to determine that the TSF modules are identified as
 9760 either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

9761 The purpose of designating each module (according to the role a particular module plays in the
 9762 enforcement of the SFRs) is to allow developers to provide less information about the parts of the
 9763 TSF that have little role in security. It is always permissible for the developer to provide more
 9764 information or detail than the requirements demand, as might occur when the information has
 9765 been gathered outside the evaluation context. In such cases the developer must still designate the
 9766 modules as either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

9767 The accuracy of these designations is continuously reviewed as the evaluation progresses. The
 9768 concern is the mis-designation of modules as being less important (and hence, having less
 9769 information) than is really the case. While blatant mis-designations may be immediately apparent
 9770 (e.g., designating an authentication module as anything but SFR-enforcing when User identification
 9771 (FIA_UID) is one of the SFRs being claimed), other mis-designations might not be discovered until
 9772 the TSF is better understood. The evaluator must therefore keep in mind that these designations
 9773 are the developer's initial best effort, but are subject to change. Further guidance is provided under
 9774 work unit ADV_TDS.5-16, which examines the accuracy of these designations.

9775 **ADV_TDS.5.3C The design shall identify all subsystems of the TSF.**

9776 **11.9.5.4.4 Work unit ADV_TDS.5-4**

9777 The evaluator **shall examine** the TOE design to determine that all subsystems of the TSF are
 9778 identified.

9779 If the design is presented solely in terms of modules, then subsystems in these requirements are
 9780 equivalent to modules and the activity should be performed at the module level.

9781 In work unit ADV_TDS.5-1 all of the subsystems of the TOE were identified, and a determination
 9782 made that the non-TSF subsystems were correctly characterised. Building on that work, the
 9783 subsystems that were not characterised as non-TSF subsystems should be precisely identified. The
 9784 evaluator determines that, of the hardware and software installed and configured according to the
 9785 Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one
 9786 that is part of the TSF, or one that is not.

9787 **ADV_TDS.5.4C The design shall provide a semiformal description of each subsystem of 9788 the TSF, supported by informal, explanatory text where appropriate.**

9789 **11.9.5.4.5 Work unit ADV_TDS.5-5**

9790 The evaluator **shall examine** the TDS documentation to determine that the semiformal notation
 9791 used for describing the subsystems, modules and their interfaces is defined or referenced.

9792 A semiformal notation can be either defined by the sponsor or a corresponding standard be
 9793 referenced. The evaluator should provide a mapping of security functions and their interfaces
 9794 outlining in what part of the documentation a function or interface is semiformal described and

- 9795 what notation is used. The evaluator examines all semiformal notations used to make sure that
 9796 they are of a semiformal style and to justify the appropriateness of the manner how the semiformal
 9797 notations are used for the TOE.
- 9798 The evaluator is reminded that a semi-formal presentation is characterised by a standardised
 9799 format with a well-defined syntax that reduces ambiguity that may occur in informal presentations.
 9800 The syntax of all semiformal notations used in the functional specification shall be defined or a
 9801 corresponding standard be referenced. The evaluator verifies that the semiformal notations used
 9802 for expressing the functional specification are capable of expressing features relevant to security.
 9803 In order to determine this, the evaluator can refer to the SFR and compare the TSF security
 9804 features stated in the ST and those described in the FSP using the semiformal notations.
- 9805 Note that ADV_TDS.5.7C requires the module description to be semiformal. This work unit
 9806 therefore applies also to that description.
- 9807 **11.9.5.4.6 Work unit ADV_TDS.5-6**
- 9808 The evaluator ***shall examine*** the TOE design to determine that each subsystem of the TSF
 9809 describes its role in the enforcement of SFRs described in the ST.
- 9810 If the design is presented solely in terms of modules, then this work unit will be considered
 9811 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
 9812 evaluator is necessary in this case.
- 9813 On systems that are complex enough to warrant a subsystem-level description of the TSF in
 9814 addition to the modular description, the goal of the subsystem-level description is to give the
 9815 evaluator context for the modular description that follows. Therefore, the evaluator ensures that
 9816 the subsystem-level description contains a description of how the security functional requirements
 9817 are achieved in the design, but at a level of abstraction above the modular description. This
 9818 description should discuss the mechanisms used at a level that is aligned with the module
 9819 description; this will provide the evaluators the road map needed to intelligently assess the
 9820 information contained in the module description. A well-written set of subsystem descriptions will
 9821 help guide the evaluator in determining the modules that are most important to examine, thus
 9822 focusing the evaluation activity on the portions of the TSF that have the most relevance with
 9823 respect to the enforcement of the SFRs.
- 9824 The evaluator ensures that all subsystems of the TSF have a description. While the description
 9825 should focus on the role that the subsystem plays in enforcing or supporting the implementation of
 9826 the SFRs, enough information must be present so that a context for understanding the SFR-related
 9827 functionality is provided.
- 9828 **11.9.5.4.7 Work unit ADV_TDS.5-7**
- 9829 The evaluator ***shall examine*** the TOE design to determine that each SFR-non-interfering
 9830 subsystem of the TSF is described such that the evaluator can determine that the subsystem is SFR-
 9831 non-interfering.
- 9832 If the design is presented solely in terms of modules, then this work unit will be considered
 9833 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
 9834 evaluator is necessary in this case.
- 9835 An SFR-non-interfering subsystem is one on which the SFR-enforcing and SFR-supporting
 9836 subsystems have no dependence; that is, they play no role in implementing SFR functionality.
- 9837 The evaluator ensures that all subsystems of the TSF have a description. While the description
 9838 should focus on the role that the subsystem do not plays in enforcing or supporting the
 9839 implementation of the SFRs, enough information must be present so that a context for
 9840 understanding the SFR-non-interfering functionality is provided.

9841 **ADV_TDS.5.5C** *The design shall provide a description of the interactions among all*
 9842 *subsystems of the TSF.*

9843 **11.9.5.4.8 Work unit ADV_TDS.5-8**

9844 The evaluator **shall examine** the TOE design to determine that interactions between the
 9845 subsystems of the TSF are described.

9846 If the design is presented solely in terms of modules, then this work unit will be considered
 9847 satisfied by the assessment done in subsequent work units; no explicit action on the part of the
 9848 evaluator is necessary in this case.

9849 On systems that are complex enough to warrant a subsystem-level description of the TSF in
 9850 addition to the modular description, the goal of describing the interactions between the
 9851 subsystems is to help provide the reader a better understanding of how the TSF performs its
 9852 functions. These interactions do not need to be characterised at the implementation level (e.g.,
 9853 parameters passed from one routine in a subsystem to a routine in a different subsystem; global
 9854 variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling
 9855 subsystem), but the data elements identified for a particular subsystem that are going to be used by
 9856 another subsystem need to be covered in this discussion. Any control relationships between
 9857 subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the
 9858 subsystem that actually implements these rules) should also be described.

9859 It should be noted while the developer should characterise all interactions between subsystems,
 9860 the evaluators need to use their own judgement in assessing the completeness of the description. If
 9861 the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for
 9862 instance, in examining the module-level documentation) that do not appear to be described, the
 9863 evaluator ensures that this information is provided by the developer. However, if the evaluator can
 9864 determine that interactions among a particular set of subsystems, while incompletely described by
 9865 the developer, and a complete description will not aid in understanding the overall functionality
 9866 nor security functionality provided by the TSF, then the evaluator may choose to consider the
 9867 description sufficient, and not pursue completeness for its own sake.

9868 **ADV_TDS.5.6C** *The design shall provide a mapping from the subsystems of the TSF to the*
 9869 *modules of the TSF.*

9870 **11.9.5.4.9 Work unit ADV_TDS.5-9**

9871 The evaluator **shall examine** the TOE design to determine that the mapping between the
 9872 subsystems of the TSF and the modules of the TSF is complete.

9873 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

9874 For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition
 9875 to the modular description, the developer provides a simple mapping showing how the modules of
 9876 the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their
 9877 module-level assessment. To determine completeness, the evaluator examines each mapping and
 9878 determines that all subsystems map to at least one module, and that all modules map to exactly one
 9879 subsystem.

9880 **11.9.5.4.10 Work unit ADV_TDS.5-10**

9881 The evaluator **shall examine** the TOE design to determine that the mapping between the
 9882 subsystems of the TSF to the modules of the TSF is accurate.

9883 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

For TOEs that are complex enough to warrant a subsystem-level description of the TSF in addition to the modular description, the developer provides a simple mapping showing how the modules of the TSF are allocated to the subsystems. This will provide the evaluator a guide in performing their module-level assessment. The evaluator may choose to check the accuracy of the mapping in conjunction with performing other work units. An “inaccurate” mapping is one where the module is mistakenly associated with a subsystem where its functions are not used within the subsystem. Because the mapping is intended to be a guide supporting more detailed analysis, the evaluator is cautioned to apply appropriate effort to this work unit. Expending extensive evaluator resources verifying the accuracy of the mapping is not necessary. Inaccuracies that lead to misunderstandings related to the design that are uncovered as part of this or other work units are the ones that should be associated with this work unit and corrected.

ADV_TDS.5.7C *The design shall provide a semiformal description of each module in terms of its purpose, interaction, interfaces, return values from those interfaces, and called interfaces to other modules, supported by informal, explanatory text where appropriate.*

11.9.5.4.11 Work unit ADV_TDS.5-11

The evaluator **shall examine** the TOE design to determine that the semiformal description of the purpose of each module, and its relationship with other modules is complete and accurate.

The developer may designate modules as SFR-enforcing, SFR-supporting, and SFR non-interfering, but these “tags” are used only to describe the amount and type of information the developer must provide, and can be used to limit the amount of information the developer has to develop if their engineering process does not produce the documentation required. Whether the modules have been categorised by the developer or not, it is the evaluator's responsibility to determine that the modules have the appropriate information for their role (SFR-enforcing, etc.) in the TOE, and to obtain the appropriate information from the developer should the developer fail to provide the required information for a particular module.

The purpose of a module provides a description indicating what function the module is fulfilling. A word of caution to the evaluator is in order. The focus of this work unit should be to provide the evaluator an understanding of how the module works so that determinations can be made about the soundness of the implementation of the SFRs, as well as to support architectural analysis performed for ADV_ARC subsystems. As long as the evaluator has a sound understanding of the module's operation, and its relationship to other modules and the TOE as a whole, the evaluator should consider the objective of the work achieved and not engage in a documentation exercise for the developer (by requiring, for example, a complete algorithmic description for a self-evident implementation representation).

Because the modules are at such a low level, it may be difficult to determine completeness and accuracy impacts from other documentation, such as operational user guidance, the functional specification, the TSF internals, or the security architecture description. However, the evaluator uses the information present in those documents to the extent possible to help ensure that the purpose is accurately and completely described. This analysis can be aided by the analysis performed for the work units for the ADV_TDS.5.8C element, which maps the TSFI in the functional specification to the modules of the TSF.

11.9.5.4.12 Work unit ADV_TDS.5-12

The evaluator **shall examine** the TOE design to determine that the semiformal description of the interfaces presented by each module contain an accurate and complete description of the related parameters, the invocation conventions for each interface, and any values returned directly by the interface.

The interfaces of a module are those interfaces used by other modules as a means to invoke the operations provided, and to provide inputs to or receive outputs from the module. The purpose in

9933 the specification of these interfaces is to permit the exercise of them during testing. Inter-module
 9934 interfaces that are not SFR-related need not be specified or described, since they are not a factor in
 9935 testing. Likewise, other internal interfaces that are not a factor in traversing SFR-related paths of
 9936 execution (such as those internal paths that are fixed).

9937 SFR-related interfaces are all interfaces that are called directly or indirectly from SFR-enforcing
 9938 modules. Those interfaces need to be described with all the parameter used in such a call. This
 9939 allows the evaluator to understand the purpose of the call in the context of operation of the SFR-
 9940 enforcing modules.

9941 SFR-related interfaces are described in terms of how they are invoked, and any values that are
 9942 returned. This description would include a list of parameters, and descriptions of these parameters.
 9943 Note that global data would also be considered parameters if used by the module (either as inputs
 9944 or outputs) when invoked. If a parameter were expected to take on a set of values (e.g., a “flag”
 9945 parameter), the complete set of values the parameter could take on, that would have an effect on
 9946 module processing, would be specified. Likewise, parameters representing data structures are
 9947 described such that each field of the data structure is identified and described. Note that different
 9948 programming languages may have additional “interfaces” that would be non-obvious; an example
 9949 would be operator/function overloading in C++. This “implicit interface” in the class description
 9950 would also be described as part of the low-level TOE design. Note that although a module could
 9951 present only one interface, it is more common that a module presents a small set of related
 9952 interfaces.

9953 In terms of the assessment of parameters (inputs and outputs) to a module, any use of global data
 9954 must also be considered. A module “uses” global data if it either reads or writes the data. In order
 9955 to assure the description of such parameters (if used) is complete, the evaluator uses other
 9956 information provided about the module in the TOE design (interfaces, algorithmic description, etc.),
 9957 as well as the description of the particular set of global data assessed in work unit ADV_TDS.5-10.
 9958 For instance, the evaluator could first determine the processing the module performs by examining
 9959 its function and interfaces presented (particularly the parameters of the interfaces). They could
 9960 then check to see if the processing appears to “touch” any of the global data areas identified in the
 9961 TDS design. The evaluator then determines that, for each global data area that appears to be
 9962 “touched”, that global data area is listed as a means of input or output by the module the evaluator
 9963 is examining.

9964 Invocation conventions are a programming-reference-type description that one could use to
 9965 correctly invoke a module's interface if one were writing a program to make use of the module's
 9966 functionality through that interface. This includes necessary inputs and outputs, including any set-
 9967 up that may need to be performed with respect to global variables.

9968 Values returned through the interface refer to values that are either passed through parameters or
 9969 messages; values that the function call itself returns in the style of a “C” program function call; or
 9970 values passed through global means (such as certain error routines in *ix-style operating systems).

9971 In order to assure the description is complete, the evaluator uses other information provided about
 9972 the module in the TOE design (e.g., algorithmic description, global data used) to ensure that it
 9973 appears all data necessary for performing the functions of the module is presented to the module,
 9974 and that any values that other modules expect the module under examination to provide are
 9975 identified as being returned by the module. The evaluator determines accuracy by ensuring that
 9976 the description of the processing matches the information listed as being passed to or from an
 9977 interface.

9978 **ADV_TDS.5.8C** *The mapping shall demonstrate that all TSFIs trace to the behaviour*
 9979 *described in the TOE design that they invoke.*

9980 **11.9.5.4.13 Work unit ADV_TDS.5-13**

9981 The evaluator *shall examine* the TOE design to determine that it contains a complete and accurate
 9982 mapping from the TSFI described in the functional specification to the modules of the TSF
 9983 described in the TOE design.

9984 The modules described in the TOE design provide a description of the implementation of the TSF.
 9985 The TSFI provide a description of how the implementation is exercised. The evidence from the
 9986 developer identifies the module that is initially invoked when an operation is requested at the TSFI,
 9987 and identify the chain of modules invoked up to the module that is primarily responsible for
 9988 implementing the functionality. However, a complete call tree for each TSFI is not required for this
 9989 work unit. The cases in which more than one module would have to be identified are where there
 9990 are “entry point” modules or wrapper modules that have no functionality other than conditioning
 9991 inputs or de-multiplexing an input. Mapping to one of these modules would not provide any useful
 9992 information to the evaluator.

9993 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at
 9994 least one module. The verification of accuracy is more complex.

9995 The first aspect of accuracy is that each TSFI is mapped to a module at the TSF boundary. This
 9996 determination can be made by reviewing the module description and its interfaces/interactions.
 9997 The next aspect of accuracy is that each TSFI identifies a chain of modules between the initial
 9998 module identified and a module that is primarily responsible for implementing the function
 9999 presented at the TSF. Note that this may be the initial module, or there may be several modules,
 10000 depending on how much pre-conditioning of the inputs is done. It should be noted that one
 10001 indicator of a pre-conditioning module is that it is invoked for a large number of the TSFI, where
 10002 the TSFI are all of similar type (e.g., system call). The final aspect of accuracy is that the mapping
 10003 makes sense. For instance, mapping a TSFI dealing with access control to a module that checks
 10004 passwords is not accurate. The evaluator should again use judgement in making this determination.
 10005 The goal is that this information aids the evaluator in understanding the system and
 10006 implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the
 10007 TSF. The bulk of the assessment of whether the SFRs are described accurately by the modules is
 10008 performed in other work units.

10009 **11.9.5.4.14 Work unit ADV_TDS.5-14**

10010 The evaluator shall examine the TOE security functional requirements and the TOE design, to
 10011 determine that all ST security functional requirements are covered by the TOE design. The
 10012 evaluator may construct a map between the TOE security functional requirements and the TOE
 10013 design. This map will likely be from a functional requirement to a set of subsystems, and later to
 10014 modules. Note that this map may have to be at a level of detail below the component or even
 10015 element level of the requirements, because of operations (assignments, refinements, selections)
 10016 performed on the functional requirement by the ST author.

10017 For example, the FDP_ACC.1 Subset access control component contains an element with
 10018 assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 Subset access
 10019 control assignment, and these ten rules were implemented in specific places within fifteen
 10020 modules, it would be inadequate for the evaluator to map FDP_ACC.1 Subset access control to
 10021 one subsystem and claim the work unit had been completed. Instead, the evaluator would map
 10022 FDP_ACC.1 Subset access control (rule 1) to modules x, y and z of subsystem A; FDP_ACC.1
 10023 Subset access control (rule 2) to x, p, and q of subsystem A; etc.

10024 **11.9.5.4.15 Work unit ADV_TDS.5-15**

10025 The evaluator shall examine the TOE design to determine that it is an accurate instantiation of all
 10026 security functional requirements.

10027 6 The evaluator may construct a map between the TOE security functional
10028 requirements and the TOE design. This map will likely be from a functional
10029 requirement to a set of subsystems and modules. Note that this map may have to
10030 be at a level of detail below the component or even element level of the
10031 requirements, because of operations (assignments, refinements, selections)
10032 performed on the functional requirement by the ST author.

10033 7 As an example, if the ST requirements specified a role-based access control
10034 mechanism, the evaluator would first identify the subsystems, and modules that
10035 contribute to this mechanism's implementation. This could be done by in-depth
10036 knowledge or understanding of the TOE design or by work done in the previous
10037 work unit. Note that this trace is only to identify the subsystems, and modules, and
10038 is not the complete analysis.

10039 8 The next step would be to understand what mechanism the subsystems, and
10040 modules implemented. For instance, if the design described an implementation of
10041 access control based on UNIX-style protection bits, the design would not be an
10042 accurate instantiation of those access control requirements present in the ST
10043 example used above. If the evaluator could not determine that the mechanism was
10044 accurately implemented because of a lack of detail, the evaluator would have to
10045 assess whether all of the SFR-enforcing subsystems and modules have been
10046 identified, or if adequate detail had been provided for those subsystems and
10047 modules.

10048

10049 **11.9.6 Evaluation of sub-activity (ADV_TDS.6)**

10050 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

10051 **12 Class AGD: Guidance documents**

10052 **12.1 Introduction**

10053 The purpose of the guidance document activity is to judge the adequacy of the documentation
10054 describing how the user can handle the TOE in a secure manner. Such documentation should take
10055 into account the various types of users (e.g. those who accept, install, administrate or operate the
10056 TOE) whose incorrect actions could adversely affect the security of the TOE or of their own data.

10057 The guidance documents class is subdivided into two families which are concerned firstly with the
10058 preparative procedures (all that has to be done to transform the delivered TOE into its evaluated
10059 configuration in the environment as described in the ST, i.e. accepting and installing the TOE) and
10060 secondly with the operational user guidance (all that has to be done during the operation of the
10061 TOE in its evaluated configuration, i.e. operation and administration).

10062 **12.2 Application notes**

10063 The guidance documents activity applies to those functions and interfaces which are related to the
10064 security of the TOE. The secure configuration of the TOE is described in the ST.

10065 **12.3 Operational user guidance (AGD_OPE)**

10066 **12.3.1 Evaluation of sub-activity (AGD_OPE.1)**

10067 **12.3.1.1 Objectives**

10068 The objectives of this sub-activity are to determine whether the user guidance describes for each
10069 user role the security functionality and interfaces provided by the TSF, provides instructions and
10070 guidelines for the secure use of the TOE, addresses secure procedures for all modes of operation,
10071 facilitates prevention and detection of insecure TOE states, or whether it is misleading or
10072 unreasonable.

10073 **12.3.1.2 Input**

10074 The evaluation evidence for this sub-activity is:

- 10075 a) the ST;
- 10076 b) the functional specification;
- 10077 c) the TOE design, if applicable;
- 10078 d) the user guidance;

10079 **12.3.1.3 Action AGD_OPE.1.1E**

10080 ISO/IEC 15408-3 AGD_OPE.1.1C: *The operational user guidance shall describe, for each user role, the*
10081 *user-accessible functions and privileges that should be controlled in a secure processing environment,*
10082 *including appropriate warnings.*

10083 **12.3.1.3.1 Work unit AGD_OPE.1-1**

10084 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each
10085 user role, the user-accessible functions and privileges that should be controlled in a secure
10086 processing environment, including appropriate warnings.

10087 The configuration of the TOE may allow different user roles to have dissimilar privileges in making
10088 use of the different functions of the TOE. This means that some users are authorised to perform
10089 certain functions, while other users may not be so authorised. These functions and privileges
10090 should be described, for each user role, by the user guidance.

10091 The user guidance identifies, for each user role, the functions and privileges that must be
10092 controlled, the types of commands required for them, and the reasons for such commands. The
10093 user guidance should contain warnings regarding the use of these functions and privileges.
10094 Warnings should address expected effects, possible side effects, and possible interactions with
10095 other functions and privileges.

10096 ISO/IEC 15408-3 AGD_OPE.1.2C: *The operational user guidance shall describe, for each user role,*
10097 *how to use the available interfaces provided by the TOE in a secure manner.*

10098 **12.3.1.3.2 Work unit AGD_OPE.1-2**

10099 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each
10100 user role, the secure use of the available interfaces provided by the TOE.

10101 The user guidance should provide advice regarding effective use of the TSF (e.g. reviewing
10102 password composition practises, suggested frequency of user file backups, discussion on the effects
10103 of changing user access privileges).

10104 ISO/IEC 15408-3 AGD_OPE.1.3C: *The operational user guidance shall describe, for each user role, the*
 10105 *available functions and interfaces, in particular all security parameters under the control of the user,*
 10106 *indicating secure values as appropriate.*

10107 **12.3.1.3.3 Work unit AGD_OPE.1-3**

10108 The evaluator **shall examine** the operational user guidance to determine that it describes, for each
 10109 user role, the available security functionality and interfaces, in particular all security parameters
 10110 under the control of the user, indicating secure values as appropriate.

10111 The user guidance should contain an overview of the security functionality that is visible at the
 10112 user interfaces.

10113 The user guidance should identify and describe the purpose, behaviour, and interrelationships of
 10114 the security interfaces and functionality.

10115 For each user-accessible interface, the user guidance should:

10116 a) describe the method(s) by which the interface is invoked (e.g. command-line,
 10117 programming-language system call, menu selection, command button);

10118 b) describe the parameters to be set by the user, their particular purposes, valid and default
 10119 values, and secure and insecure use settings of such parameters, both individually or in
 10120 combination;

10121 c) describe the immediate TSF response, message, or code returned.

10122 The evaluator should consider the functional specification and the ST to determine that the TSF
 10123 described in these documents is consistent to the operational user guidance. The evaluator has to
 10124 ensure that the operational user guidance is complete to allow the secure use through the TSFI
 10125 available to all types of human users. The evaluator may, as an aid, prepare an informal mapping
 10126 between the guidance and these documents. Any omissions in this mapping may indicate
 10127 incompleteness.

10128 ISO/IEC 15408-3 AGD_OPE.1.4C: *The operational user guidance shall, for each user role, clearly*
 10129 *present each type of security-relevant event relative to the user-accessible functions that need to be*
 10130 *performed, including changing the security characteristics of entities under the control of the TSF.*

10131 **12.3.1.3.4 Work unit AGD_OPE.1-4**

10132 The evaluator **shall examine** the operational user guidance to determine that it describes, for each
 10133 user role, each type of security-relevant event relative to the user functions that need to be
 10134 performed, including changing the security characteristics of entities under the control of the TSF
 10135 and operation following failure or operational error.

10136 All types of security-relevant events are detailed for each user role, such that each user knows
 10137 what events may occur and what action (if any) they may have to take in order to maintain security.
 10138 Security-relevant events that may occur during operation of the TOE (e.g. audit trail overflow,
 10139 system crash, updates to user records, such as when a user account is removed when the user
 10140 leaves the organisation) are adequately defined to allow user intervention to maintain secure
 10141 operation.

10142 ISO/IEC 15408-3 AGD_OPE.1.5C: *The operational user guidance shall identify all possible modes of*
 10143 *operation of the TOE (including operation following failure or operational error), their consequences*
 10144 *and implications for maintaining secure operation.*

10145 **12.3.1.3.5 Work unit AGD_OPE.1-5**

10146 The evaluator **shall examine** the operational user guidance and other evaluation evidence to
 10147 determine that the guidance identifies all possible modes of operation of the TOE (including, if
 10148 applicable, operation following failure or operational error), their consequences and implications
 10149 for maintaining secure operation.

10150 Other evaluation evidence, particularly the functional specification, provide an information source
 10151 that the evaluator should use to determine that the guidance contains sufficient guidance
 10152 information.

10153 If test documentation is included in the assurance package, then the information provided in this
 10154 evidence can also be used to determine that the guidance contains sufficient guidance
 10155 documentation. The detail provided in the test steps can be used to confirm that the guidance
 10156 provided is sufficient for the use and administration of the TOE.

10157 The evaluator should focus on a single human visible TSFI at a time, comparing the guidance for
 10158 securely using the TSFI with other evaluation evidence, to determine that the guidance related to
 10159 the TSFI is sufficient for the secure usage (i.e. consistent with the SFRs) of that TSFI. The evaluator
 10160 should also consider the relationships between interfaces, searching for potential conflicts.

10161 ISO/IEC 15408-3 AGD_OPE.1.6C: *The operational user guidance shall, for each user role, describe the*
 10162 *security measures to be followed in order to fulfil the security objectives for the operational*
 10163 *environment as described in the ST.*

10164 **12.3.1.3.6 Work unit AGD_OPE.1-6**

10165 The evaluator **shall examine** the operational user guidance to determine that it describes, for each
 10166 user role, the security measures to be followed in order to fulfil the security objectives for the
 10167 operational environment as described in the ST.

10168 The evaluator analyses the security objectives for the operational environment in the ST and
 10169 determines that for each user role, the relevant security measures are described appropriately in
 10170 the user guidance.

10171 The security measures described in the user guidance should include all relevant external
 10172 procedural, physical, personnel and connectivity measures.

10173 Note that those measures relevant for secure installation of the TOE are examined in Preparative
 10174 procedures (AGD_PRE).

10175 ISO/IEC 15408-3 AGD_OPE.1.7C: *The operational user guidance shall be clear and reasonable.*

10176 **12.3.1.3.7 Work unit AGD_OPE.1-7**

10177 The evaluator **shall examine** the operational user guidance to determine that it is clear.

10178 The guidance is unclear if it can reasonably be misconstrued by an administrator or user, and used
 10179 in a way detrimental to the TOE, or to the security provided by the TOE.

10180 **12.3.1.3.8 Work unit AGD_OPE.1-8**

10181 The evaluator **shall examine** the operational user guidance to determine that it is reasonable.

10182 The guidance is unreasonable if it makes demands on the TOE's usage or operational environment
 10183 that are inconsistent with the ST or unduly onerous to maintain security.

10184 **12.4 Preparative procedures (AGD_PRE)**

10185 **12.4.1 Evaluation of sub-activity (AGD_PRE.1)**

10186 **12.4.1.1 Objectives**

10187 The objective of this sub-activity is to determine whether the procedures and steps for the secure
10188 preparation of the TOE have been documented and result in a secure configuration.

10189 **12.4.1.2 Input**

10190 The evaluation evidence for this sub-activity is:

- 10191 a) the ST;
- 10192 b) the TOE including its preparative procedures;
- 10193 c) the description of developer's delivery procedures, if applicable;

10194 **12.4.1.3 Application notes**

10195 The preparative procedures refer to all acceptance and installation procedures, that are necessary
10196 to progress the TOE to the secure configuration as described in the ST.

10197 **12.4.1.4 Action AGD_PRE.1.1E**

10198 ISO/IEC 15408-3 AGD_PRE.1.1C: *The preparative procedures shall describe all the steps necessary*
10199 *for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.*

10200 **12.4.1.4.1 Work unit AGD_PRE.1-1**

10201 The evaluator ***shall examine*** the provided acceptance procedures to determine that they describe
10202 the steps necessary for secure acceptance of the TOE in accordance with the developer's delivery
10203 procedures.

10204 If it is not anticipated by the developer's delivery procedures that acceptance procedures will or
10205 can be applied, this work unit is not applicable, and is therefore considered to be satisfied.

10206 The acceptance procedures should include as a minimum, that the user has to check that all parts
10207 of the TOE as indicated in the ST have been delivered in the correct version.

10208 The acceptance procedures should reflect the steps the user has to perform in order to accept the
10209 delivered TOE that are implied by the developer's delivery procedures.

10210 The acceptance procedures should provide detailed information about the following, if applicable:

- 10211 a) making sure that the delivered TOE is the complete evaluated instance;
- 10212 b) detecting modification/masquerading of the delivered TOE.

10213 ISO/IEC 15408-3 AGD_PRE.1.2C: *The preparative procedures shall describe all the steps necessary*
10214 *for secure installation of the TOE and for the secure preparation of the operational environment in*
10215 *accordance with the security objectives for the operational environment as described in the ST.*

10216 **12.4.1.4.2 Work unit AGD_PRE.1-2**

10217 The evaluator ***shall examine*** the provided installation procedures to determine that they describe
 10218 the steps necessary for secure installation of the TOE and the secure preparation of the operational
 10219 environment in accordance with the security objectives in the ST.

10220 If it is not anticipated that installation procedures will or can be applied (e.g. because the TOE may
 10221 already be delivered in an operational state), this work unit is not applicable, and is therefore
 10222 considered to be satisfied.

10223 The installation procedures should provide detailed information about the following, if applicable:

- 10224 a) minimum system requirements for secure installation;
- 10225 b) requirements for the operational environment in accordance with the security objectives
 10226 provided by the ST;
- 10227 c) the steps the user has to perform in order to get to an operational TOE being
 10228 commensurate with its evaluated configuration. Such a description shall include - for
 10229 each step - a clear scheme for the decision on the next step depended on success, failure
 10230 or problems at the current step;
- 10231 d) changing the installation specific security characteristics of entities under the control of
 10232 the TSF (for example parameters, settings, passwords);
- 10233 e) handling exceptions and problems.

10234 **12.4.1.5 Action AGD_PRE.1.2E**10235 **12.4.1.5.1 Work unit AGD_PRE.1-3**

10236 The evaluator ***shall perform*** all user procedures necessary to prepare the TOE to determine that
 10237 the TOE and its operational environment can be prepared securely using only the supplied
 10238 preparative procedures.

10239 Preparation requires the evaluator to advance the TOE from a deliverable state to the state in
 10240 which it is operational, including acceptance and installation of the TOE, and enforcing the SFRs
 10241 consistent with the security objectives for the TOE specified in the ST.

10242 The evaluator should follow only the developer's procedures and may perform the activities that
 10243 customers are usually expected to perform to accept and install the TOE, using the supplied
 10244 preparative procedures only. Any difficulties encountered during such an exercise may be
 10245 indicative of incomplete, unclear or unreasonable guidance.

10246 **12.4.1.6 This work unit may be performed in conjunction with the evaluation activities**
 10247 **under Objectives**

10248 **12.4.1.6** The objective of this sub-activity is to determine whether the developer correctly performed and
 10249 documented the tests in the test documentation and to ensure that testing is structured such as to
 10250 avoid circular arguments about the correctness of the interfaces being tested.

10251 **12.4.1.6 Input**

10252 **12.4.1.6** The evaluation evidence for this sub-activity is:

10253 **12.4.1.6** the ST;

10254 **12.4.1.6** the functional specification;

- 10255 **12.4.1.6** the test documentation.
- 10256 **12.4.1.6 Application notes**
- 10257 **12.4.1.6** Although the test procedures may state pre-requisite initial test conditions in terms of
 10258 ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering,
 10259 which provides this rationale, is an important factor in determining the adequacy of testing, as
 10260 there is a possibility of faults being concealed by the ordering of tests.
- 10261 **12.4.1.6 Action ATE_FUN.2.1E**
- 10262 **12.4.1.6** ISO/IEC 15408-3 ATE_FUN.2.1C ***The test documentation shall consist of test plans,***
 10263 ***expected test results and actual test results.***
- 10264 **12.4.1.6 Work unit ATE_FUN.2-1**
- 10265 **12.4.1.6** The evaluator ***shall check that the test documentation includes test plans, expected test***
 10266 ***results and actual test results.***
- 10267 **12.4.1.6** The evaluator checks that test plans, expected tests results and actual test results are
 10268 included in the test documentation.
- 10269 **12.4.1.6** ISO/IEC 15408-3 ATE_FUN.2.2C ***The test plans shall identify the tests to be performed***
 10270 ***and describe the scenarios for performing each test. These scenarios shall include any ordering***
 10271 ***dependencies on the results of other tests.***
- 10272 **12.4.1.6 Work unit ATE_FUN.2-2**
- 10273 **12.4.1.6** The evaluator ***shall examine the test plan to determine that it describes the scenarios***
 10274 ***for performing each test.***
- 10275 **12.4.1.6** The evaluator determines that the test plan provides information about the test
 10276 configuration being used: both on the configuration of the TOE and on any test equipment being
 10277 used. This information should be detailed enough to ensure that the test configuration is
 10278 reproducible.
- 10279 **12.4.1.6** The evaluator also determines that the test plan provides information about how to
 10280 execute the test: any necessary automated set-up procedures (and whether they require privilege
 10281 to run), inputs to be applied, how these inputs are applied, how output is obtained, any
 10282 automated clean-up procedures (and whether they require privilege to run), etc. This
 10283 information should be detailed enough to ensure that the test is reproducible.
- 10284 **12.4.1.6** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 10285 **12.4.1.6 Work unit ATE_FUN.2-3**
- 10286 **12.4.1.6** The evaluator ***shall examine the test plan to determine that the TOE test configuration***
 10287 ***is consistent with the ST.***
- 10288 **12.4.1.6** The TOE referred to in the developer's test plan should have the same unique reference
 10289 as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
 10290 introduction.
- 10291 **12.4.1.6** It is possible for the ST to specify more than one configuration for evaluation. The
 10292 evaluator verifies that all test configurations identified in the developer test documentation are
 10293 consistent with the ST. For example, the ST might define configuration options that must be set,
 10294 which could have an impact upon what constitutes the TOE by including or excluding additional
 10295 portions. The evaluator verifies that all such variations of the TOE are considered.

- 10296 **12.4.1.6** The evaluator should consider the security objectives for the operational environment
 10297 described in the ST that may apply to the test environment. There may be some objectives for the
 10298 operational environment that do not apply to the test environment. For example, an objective
 10299 about user clearances may not apply; however, an objective about a single point of connection to
 10300 a network would apply.
- 10301 **12.4.1.6** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 10302 **12.4.1.6** If this work unit is applied to a component TOE that might be used/integrated in a
 10303 composed TOE (see Class ACO: Composition), the following will apply. In the instances that the
 10304 component TOE under evaluation depends on other components in the operational environment
 10305 to support their operation, the developer may wish to consider using the other component(s)
 10306 that will be used in the composed TOE to fulfil the requirements of the operational environment
 10307 as one of the test configurations. This will reduce the amount an additional testing that will be
 10308 required for the composed TOE evaluation.
- 10309 **12.4.1.6 Work unit ATE_FUN.2-4**
- 10310 **12.4.1.6** The evaluator *shall examine the test plans to determine that sufficient instructions are*
 10311 *provided for any ordering dependencies.*
- 10312 **12.4.1.6** Some steps may have to be performed to establish initial conditions. For example, user
 10313 accounts need to be added before they can be deleted. An example of ordering dependencies on
 10314 the results of other tests is the need to perform actions in a test that will result in the generation
 10315 of audit records, before performing a test to consider the searching and sorting of those audit
 10316 records. Another example of an ordering dependency would be where one test case generates a
 10317 file of data to be used as input for another test case.
- 10318 **12.4.1.6** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 10319 **12.4.1.6 ATE_FUN.2.3C** *The expected test results shall show the anticipated outputs*
 10320 *from a successful execution of the tests.*
- 10321 **12.4.1.6 Work unit ATE_FUN.2-5**
- 10322 **12.4.1.6** The evaluator *shall examine the test documentation to determine that all expected*
 10323 *tests results are included.*
- 10324 **12.4.1.6** The expected test results are needed to determine whether or not a test has been
 10325 successfully performed. Expected test results are sufficient if they are unambiguous and
 10326 consistent with expected behaviour given the testing approach.
- 10327 **12.4.1.6** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 10328 **12.4.1.6 ATE_FUN.2.4C** *The actual test results shall be consistent with the expected*
 10329 *test results.*
- 10330 **12.4.1.6 Work unit ATE_FUN.2-6**
- 10331 **12.4.1.6** The evaluator *shall check that the actual test results in the test documentation are*
 10332 *consistent with the expected test results in the test documentation.*
- 10333 **12.4.1.6** A comparison of the actual and expected test results provided by the developer will
 10334 reveal any inconsistencies between the results. It may be that a direct comparison of actual
 10335 results cannot be made until some data reduction or synthesis has been first performed. In such
 10336 cases, the developer's test documentation should describe the process to reduce or synthesise the
 10337 actual data.

- 10338 **12.4.1.6** For example, the developer may need to test the contents of a message buffer after a
 10339 network connection has occurred to determine the contents of the buffer. The message buffer will
 10340 contain a binary number. This binary number would have to be converted to another form of data
 10341 representation in order to make the test more meaningful. The conversion of this binary
 10342 representation of data into a higher-level representation will have to be described by the
 10343 developer in enough detail to allow an evaluator to perform the conversion process (i.e.
 10344 synchronous or asynchronous transmission, number of stop bits, parity, etc.).
- 10345 **12.4.1.6** It should be noted that the description of the process used to reduce or synthesise the
 10346 actual data is used by the evaluator not to actually perform the necessary modification but to
 10347 assess whether this process is correct. It is up to the developer to transform the expected test
 10348 results into a format that allows an easy comparison with the actual test results.
- 10349 **12.4.1.6** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 10350 **12.4.1.6 Work unit ATE_FUN.2-7**
- 10351 **12.4.1.6** The evaluator **shall report the developer testing effort, outlining the testing approach,**
 10352 configuration, depth and results.
- 10353 **12.4.1.6** The developer testing information recorded in the ETR allows the evaluator to convey
 10354 the overall testing approach and effort expended on the testing of the TOE by the developer. The
 10355 intent of providing this information is to give a meaningful overview of the developer testing
 10356 effort. It is not intended that the information regarding developer testing in the ETR be an exact
 10357 reproduction of specific test steps or results of individual tests. The intention is to provide
 10358 enough detail to allow other evaluators and evaluation authorities to gain some insight about the
 10359 developer's testing approach, amount of testing performed, TOE test configurations, and the
 10360 overall results of the developer testing.
- 10361 **12.4.1.6** Information that would typically be found in the ETR section regarding the developer
 10362 testing effort is:
- 10363 **12.4.1.6** TOE test configurations. The particular configurations of the TOE that were tested,
 10364 including whether any privileged code was required to set up the test or clean up afterwards;
- 10365 **12.4.1.6** testing approach. An account of the overall developer testing strategy employed;
- 10366 **12.4.1.6** testing results. A description of the overall developer testing results.
- 10367 **12.4.1.6** This list is by no means exhaustive and is only intended to provide some context as to
 10368 the type of information that should be present in the ETR concerning the developer testing effort.
- 10369 **12.4.1.6 ATE_FUN.2.5C *The test documentation shall include an analysis of the test***
 10370 ***procedure ordering dependencies.***
- 10371 **12.4.1.6 Work unit ATE_FUN.2-8**
- 10372 **12.4.1.6** The evaluator ***shall examine the analysis of the test procedure ordering dependencies***
 10373 to determine that a sufficient justification for the chosen ordering of test cases is given.
- 10374 **12.4.1.6** Usually the evaluator will generate a table of all cases, where the test documentation
 10375 requires a certain ordering of the tests and will then examine if sufficient justification is given in
 10376 any case, why testing in this ordering is adequate and sufficient.
- 10377 **12.4.1.6** As an example we assume that the TSF provide a random number generator, which
 10378 needs to be initialised (for example with an adequate seed) before random numbers of a specified
 10379 quality can be generated. In this case the evaluator will consider the following question:

- 10380 **12.4.1.6** Does the test documentation only describe an ordering of tests, where the initialisation
10381 is done before calling the function to generate a random number?
- 10382 **12.4.1.6** In this case the justification needs to show, why the developer expects, that in the
10383 intended environment of the TOE the random number function will not be called without
10384 initialisation of the random number generator.
- 10385 **12.4.1.6** If for example the user guidance documentation includes a clear instruction that the
10386 random number generator needs to be initialised adequately before being called, this may be
10387 considered adequate as a justification. (note that the question if it can be plausibly assumed that
10388 users will follow such instruction is covered by the evaluation activities for the classes ASE and
10389 AGD and needs not to be re-examined here.)
- 10390 **12.4.1.6** If, on the other hand, the TOE provides an authentication protocol, which implicitly uses
10391 random numbers provided by the random number generator, and an attacker can therefore "call"
10392 the random number generator implicitly by simply trying to authenticate himself, and if neither
10393 the TOE nor the environment prevent an attacker from doing this even before the random
10394 number generator is initialised, a test case needs to show, what happens in such situation.
- 10395 **12.4.1.6** If, for example, instead of returning a "bad" random number, the random number
10396 function would return an error, when called without proper initialisation, it would be much
10397 better to include a test showing this secure behaviour instead of trying to justify why the
10398 functions are only tested in the usual order.
- 10399 **12.4.1.6** Note: Of course even without ATE_FUN.2 an evaluator would be expected to look for
10400 potential vulnerabilities like the one described above. However, ATE_FUN.2.5C adds assurance by
10401 requiring the developer to give a systematic justification, why their **chosen order of test cases**
10402 doesn't hide such potential failures of security functions.
- 10403 **12.4.1.6** Independent testing (ATE_IND).
- 10404 If it is known that the TOE will be used as a dependent component for a composed TOE evaluation,
10405 then the evaluator should ensure that the operational environment is satisfied by the base
10406 component used in the composed TOE.
- 10407 **13 Class ALC: Life-cycle support**
- 10408 **13.1 Introduction**
- 10409 The purpose of the life-cycle support activity is to determine the adequacy of the security
10410 procedures that the developer uses during the development and maintenance of the TOE. These
10411 procedures include the life-cycle model used by the developer, the configuration management, the
10412 security measures used throughout TOE development, the tools used by the developer throughout
10413 the life-cycle of the TOE, the handling of security flaws, and the delivery activity.
- 10414 Poorly controlled development and maintenance of the TOE can result in vulnerabilities in the
10415 implementation. Conformance to a defined life-cycle model can help to improve controls in this
10416 area. A measurable life-cycle model used for the TOE can remove ambiguity in assessing the
10417 development progress of the TOE.
- 10418 The purpose of the configuration management activity is to assist the consumer in identifying the
10419 evaluated TOE, to ensure that configuration items are uniquely identified, and the adequacy of the
10420 procedures that are used by the developer to control and track changes that are made to the TOE.
10421 This includes details on what changes are tracked, how potential changes are incorporated, and the
10422 degree to which automation is used to reduce the scope for error.

10423 Developer security procedures are intended to protect the TOE and its associated design
10424 information from interference or disclosure. Interference in the development process may allow
10425 the deliberate introduction of vulnerabilities. Disclosure of design information may allow
10426 vulnerabilities to be more easily exploited. The adequacy of the procedures will depend on the
10427 nature of the TOE and the development process.

10428 The use of well-defined development tools and the application of implementation standards by the
10429 developer and by third parties involved in the development process help to ensure that
10430 vulnerabilities are not inadvertently introduced during refinement.

10431 The flaw remediation activity is intended to track security flaws, to identify corrective actions, and
10432 to distribute the corrective action information to TOE users.

10433 The purpose of the delivery activity is to judge the adequacy of the documentation of the
10434 procedures used to ensure that the TOE is delivered to the consumer without modification.

10435 **13.2 CM capabilities (ALC_CMC)**

10436 **13.2.1 Evaluation of sub-activity (ALC_CMC.1)**

10437 **13.2.1.1 Objectives**

10438 The objectives of this sub-activity are to determine whether the developer has clearly identified the
10439 TOE.

10440 **13.2.1.2 Input**

10441 The evaluation evidence for this sub-activity is:

10442 a) the ST;

10443 b) the TOE suitable for testing.

10444 **13.2.1.3 Action ALC_CMC.1.1E**

10445 ISO/IEC 15408-3 ALC_CMC.1.1C: *The TOE shall be labelled with its unique reference.*

10446 **13.2.1.3.1 Work unit ALC_CMC.1-1**

10447 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

10448 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST.
10449 This could be achieved through labelled packaging or media, or by a label displayed by the
10450 operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g.
10451 at the point of purchase or use).

10452 The TOE may provide a method by which it can be easily identified. For example, a software TOE
10453 may display its name and version number during the start up routine, or in response to a command
10454 line entry. A hardware or firmware TOE may be identified by a part number physically stamped on
10455 the TOE.

10456 Alternatively, the unique reference provided for the TOE may be the combination of the unique
10457 reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

10458 **13.2.1.3.2 Work unit ALC_CMC.1-2**

10459 The evaluator ***shall check*** that the TOE references used are consistent.

10460 If the TOE is labelled more than once then the labels have to be consistent. For example, it should
 10461 be possible to relate any labelled guidance documentation supplied as part of the TOE to the
 10462 evaluated operational TOE. This ensures that consumers can be confident that they have purchased
 10463 the evaluated version of the TOE, that they have installed this version, and that they have the
 10464 correct version of the guidance to operate the TOE in accordance with its ST.

10465 The evaluator also verifies that the TOE reference is consistent with the ST.

10466 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will
 10467 not be labelled with its unique (composite) reference, but only the individual components will be
 10468 labelled with their appropriate TOE reference. It would require further development for the IT TOE
 10469 to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed
 10470 TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain
 10471 the composite reference. However, the composed TOE ST will include the unique reference for the
 10472 composed TOE and will identify the components comprising the composed TOE through which the
 10473 consumers will be able to determine whether they have the appropriate items.

10474 **13.2.2 Evaluation of sub-activity (ALC_CMC.2)**

10475 **13.2.2.1 Objectives**

10476 The objectives of this sub-activity are to determine whether the developer uses a CM system that
 10477 uniquely identifies all configuration items.

10478 **13.2.2.2 Input**

10479 The evaluation evidence for this sub-activity is:

- 10480 a) the ST;
- 10481 b) the TOE suitable for testing;
- 10482 c) the configuration management documentation.

10483 **13.2.2.3 Application notes**

10484 This component contains an implicit evaluator action to determine that the CM system is being
 10485 used. As the requirements here are limited to identification of the TOE and provision of a
 10486 configuration list, this action is already covered by, and limited to, the existing work units. At
 10487 Evaluation of sub-activity (ALC_CMC.3) the requirements are expanded beyond these two items,
 10488 and more explicit evidence of operation is required.

10489 **13.2.2.4 Action ALC_CMC.2.1E**

10490 ISO/IEC 15408-3 ALC_CMC.2.1C: *The TOE shall be labelled with its unique reference.*

10491 **13.2.2.4.1 Work unit ALC_CMC.2-1**

10492 The evaluator **shall check** that the TOE provided for evaluation is labelled with its reference.

10493 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST.
 10494 This could be achieved through labelled packaging or media, or by a label displayed by the
 10495 operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g.
 10496 at the point of purchase or use).

10497 The TOE may provide a method by which it can be easily identified. For example, a software TOE
 10498 may display its name and version number during the start up routine, or in response to a command

- 10499 line entry. A hardware or firmware TOE may be identified by a part number physically stamped on
10500 the TOE.
- 10501 Alternatively, the unique reference provided for the TOE may be the combination of the unique
10502 reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).
- 10503 **13.2.2.4.2 Work unit ALC_CMC.2-2**
- 10504 The evaluator **shall check** that the TOE references used are consistent.
- 10505 If the TOE is labelled more than once then the labels have to be consistent. For example, it should
10506 be possible to relate any labelled guidance documentation supplied as part of the TOE to the
10507 evaluated operational TOE. This ensures that consumers can be confident that they have purchased
10508 the evaluated version of the TOE, that they have installed this version, and that they have the
10509 correct version of the guidance to operate the TOE in accordance with its ST.
- 10510 The evaluator also verifies that the TOE reference is consistent with the ST.
- 10511 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will
10512 not be labelled with its unique (composite) reference, but only the individual components will be
10513 labelled with their appropriate TOE reference. It would require further development for the IT TOE
10514 to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed
10515 TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain
10516 the composite reference. However, the composed TOE ST will include the unique reference for the
10517 composed TOE and will identify the components comprising the composed TOE through which the
10518 consumers will be able to determine whether they have the appropriate items.
- 10519 ISO/IEC 15408-3 ALC_CMC.2.2C: *The CM documentation shall describe the method used to uniquely*
10520 *identify the configuration items.*
- 10521 **13.2.2.4.3 Work unit ALC_CMC.2-3**
- 10522 The evaluator **shall examine** the method of identifying configuration items to determine that it
10523 describes how configuration items are uniquely identified.
- 10524 Procedures should describe how the status of each configuration item can be tracked throughout
10525 the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM
10526 documentation. The information included should describe:
- 10527 a) the method how each configuration item is uniquely identified, such that it is possible to
10528 track versions of the same configuration item;
- 10529 b) the method how configuration items are assigned unique identifiers and how they are
10530 entered into the CM system;
- 10531 c) the method to be used to identify superseded versions of a configuration item.
- 10532 ISO/IEC 15408-3 ALC_CMC.2.3C: *The CM system shall uniquely identify all configuration items.*
- 10533 **13.2.2.4.4 Work unit ALC_CMC.2-4**
- 10534 The evaluator **shall examine** the configuration items to determine that they are identified in a way
10535 that is consistent with the CM documentation.
- 10536 Assurance that the CM system uniquely identifies all configuration items is gained by examining
10537 the identifiers for the configuration items. For both configuration items that comprise the TOE, and
10538 drafts of configuration items that are submitted by the developer as evaluation evidence, the

10539 evaluator confirms that each configuration item possesses a unique identifier in a manner
10540 consistent with the unique identification method that is described in the CM documentation.

10541 **13.2.3 Evaluation of sub-activity (ALC_CMC.3)**

10542 **13.2.3.1 Objectives**

10543 The objectives of this sub-activity are to determine whether the developer uses a CM system that
10544 uniquely identifies all configuration items, and whether the ability to modify these items is
10545 properly controlled.

10546 **13.2.3.2 Input**

10547 The evaluation evidence for this sub-activity is:

- 10548 a) the ST;
- 10549 b) the TOE suitable for testing;
- 10550 c) the configuration management documentation.

10551 **13.2.3.3 Action ALC_CMC.3.1E**

10552 ISO/IEC 15408-3 ALC_CMC.3.1C: *The TOE shall be labelled with its unique reference.*

10553 **13.2.3.3.1 Work unit ALC_CMC.3-1**

10554 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

10555 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST.
10556 This could be achieved through labelled packaging or media, or by a label displayed by the
10557 operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g.
10558 at the point of purchase or use).

10559 The TOE may provide a method by which it can be easily identified. For example, a software TOE
10560 may display its name and version number during the start up routine, or in response to a command
10561 line entry. A hardware or firmware TOE may be identified by a part number physically stamped on
10562 the TOE.

10563 Alternatively, the unique reference provided for the TOE may be the combination of the unique
10564 reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

10565 **13.2.3.3.2 Work unit ALC_CMC.3-2**

10566 The evaluator ***shall check*** that the TOE references used are consistent.

10567 If the TOE is labelled more than once then the labels have to be consistent. For example, it should
10568 be possible to relate any labelled guidance documentation supplied as part of the TOE to the
10569 evaluated operational TOE. This ensures that consumers can be confident that they have purchased
10570 the evaluated version of the TOE, that they have installed this version, and that they have the
10571 correct version of the guidance to operate the TOE in accordance with its ST.

10572 The evaluator also verifies that the TOE reference is consistent with the ST.

10573 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will
10574 not be labelled with its unique (composite) reference, but only the individual components will be
10575 labelled with their appropriate TOE reference. It would require further development for the IT TOE
10576 to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed

10577 TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain
10578 the composite reference. However, the composed TOE ST will include the unique reference for the
10579 composed TOE and will identify the components comprising the composed TOE through which the
10580 consumers will be able to determine whether they have the appropriate items.

10581 ISO/IEC 15408-3 ALC_CMC.3.2C: *The CM documentation shall describe the method used to uniquely*
10582 *identify the configuration items.*

10583 **13.2.3.3.3 Work unit ALC_CMC.3-3**

10584 The evaluator **shall examine** the method of identifying configuration items to determine that it
10585 describes how configuration items are uniquely identified.

10586 Procedures should describe how the status of each configuration item can be tracked throughout
10587 the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM
10588 documentation. The information included should describe:

10589 a) the method how each configuration item is uniquely identified, such that it is possible to
10590 track versions of the same configuration item;

10591 b) the method how configuration items are assigned unique identifiers and how they are
10592 entered into the CM system;

10593 c) the method to be used to identify superseded versions of a configuration item.

10594 ISO/IEC 15408-3 ALC_CMC.3.3C: *The CM system shall uniquely identify all configuration items.*

10595 **13.2.3.3.4 Work unit ALC_CMC.3-4**

10596 The evaluator **shall examine** the configuration items to determine that they are identified in a way
10597 that is consistent with the CM documentation.

10598 Assurance that the CM system uniquely identifies all configuration items is gained by examining
10599 the identifiers for the configuration items. For both configuration items that comprise the TOE, and
10600 drafts of configuration items that are submitted by the developer as evaluation evidence, the
10601 evaluator confirms that each configuration item possesses a unique identifier in a manner
10602 consistent with the unique identification method that is described in the CM documentation.

10603 ISO/IEC 15408-3 ALC_CMC.3.4C: *The CM system shall provide measures such that only authorised*
10604 *changes are made to the configuration items.*

10605 **13.2.3.3.5 Work unit ALC_CMC.3-5**

10606 The evaluator **shall examine** the CM access control measures described in the CM plan to
10607 determine that they are effective in preventing unauthorised access to the configuration items.

10608 The evaluator may use a number of methods to determine that the CM access control measures are
10609 effective. For example, the evaluator may exercise the access control measures to ensure that the
10610 procedures could not be bypassed. The evaluator may use the outputs generated by the CM system
10611 procedures required by **ALC_CMC.3.8C**. The evaluator may also witness a demonstration of the CM
10612 system to ensure that the access control measures employed are operating effectively.

10613 ISO/IEC 15408-3 ALC_CMC.3.5C: *The CM documentation shall include a CM plan.*

10614 **13.2.3.3.6 Work unit ALC_CMC.3-6**

10615 The evaluator **shall check** that the CM documentation provided includes a CM plan.

10616 The CM plan needs not to be a connected document, but it is recommended that there is a single
 10617 document that describes where the various parts of the CM plan can be found. If the CM plan is no
 10618 single document, the list in the following work unit gives hints regarding which context is expected.

10619 ISO/IEC 15408-3 ALC_CMC.3.6C: *The CM plan shall describe how the CM system is used for the*
 10620 *development of the TOE.*

10621 **13.2.3.3.7 Work unit ALC_CMC.3-7**

10622 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used
 10623 for the development of the TOE.

10624 The descriptions contained in a CM plan include, if applicable:

10625 a) all activities performed in the TOE development that are subject to configuration
 10626 management procedures (e.g. creation, modification or deletion of a configuration item,
 10627 data-backup, archiving);

10628 b) which means (e.g. CM tools, forms) have to be made available;

10629 c) the usage of the CM tools: the necessary details for a user of the CM system to be able to
 10630 operate the CM tools correctly in order to maintain the integrity of the TOE;

10631 d) which other objects (development components, tools, assessment environments, etc) are
 10632 taken under CM control;

10633 e) the roles and responsibilities of individuals required to perform operations on individual
 10634 configuration items (different roles may be identified for different types of configuration
 10635 items (e.g. design documentation or source code));

10636 f) how CM instances (e.g. change control boards, interface control working groups) are
 10637 introduced and staffed;

10638 g) the description of the change management;

10639 h) the procedures that are used to ensure that only authorised individuals can make changes
 10640 to configuration items;

10641 i) the procedures that are used to ensure that concurrency problems do not occur as a
 10642 result of simultaneous changes to configuration items;

10643 j) the evidence that is generated as a result of application of the procedures. For example,
 10644 for a change to a configuration item, the CM system might record a description of the
 10645 change, accountability for the change, identification of all configuration items affected,
 10646 status (e.g. pending or completed), and date and time of the change. This might be
 10647 recorded in an audit trail of changes made or change control records;

10648 k) the approach to version control and unique referencing of TOE versions (e.g. covering the
 10649 release of patches in operating systems, and the subsequent detection of their
 10650 application).

10651 ISO/IEC 15408-3 ALC_CMC.3.7C: *The evidence shall demonstrate that all configuration items are*
 10652 *being maintained under the CM system.*

10653 **13.2.3.3.8 Work unit ALC_CMC.3-8**

10654 The evaluator ***shall check*** that the configuration items identified in the configuration list are being
 10655 maintained by the CM system.

10656 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator
10657 should check that for each type of configuration item (e.g. design documents or source code
10658 modules) contained in the configuration list there are examples of the evidence generated by the
10659 procedures described in the CM plan. In this case, the approach to sampling will depend upon the
10660 level of granularity used in the CM system to control CM items. Where, for example, 10,000 source
10661 code modules are identified in the configuration list, a different sampling strategy needs to be
10662 applied compared to the case in which there are only 5, or even 1. The emphasis of this activity
10663 should be on ensuring that the CM system is being operated correctly, rather than on the detection
10664 of any minor error.

10665 For guidance on sampling see A.2, Sampling.

10666 ISO/IEC 15408-3 ALC_CMC.3.8C: *The evidence shall demonstrate that the CM system is being*
10667 *operated in accordance with the CM plan.*

10668 **13.2.3.3.9 Work unit ALC_CMC.3-9**

10669 The evaluator **shall check** the CM documentation to ascertain that it includes the CM system
10670 records identified by the CM plan.

10671 The output produced by the CM system should provide the evidence that the evaluator needs to be
10672 confident that the CM plan is being applied, and also that all configuration items are being
10673 maintained by the CM system as required by **ALC_CMC.3.7C**. Example output could include change
10674 control forms, or configuration item access approval forms.

10675 **13.2.3.3.10 Work unit ALC_CMC.3-10**

10676 The evaluator **shall examine** the evidence to determine that the CM system is being operated in
10677 accordance with the CM plan.

10678 The evaluator should select and examine a sample of evidence covering each type of CM-relevant
10679 operation that has been performed on a configuration item (e.g. creation, modification, deletion,
10680 reversion to an earlier version) to confirm that all operations of the CM system have been carried
10681 out in line with documented procedures. The evaluator confirms that the evidence includes all the
10682 information identified for that operation in the CM plan. Examination of the evidence may require
10683 access to a CM tool that is used. The evaluator may choose to sample the evidence.

10684 For guidance on sampling see A.2, Sampling.

10685 Further confidence in the correct operation of the CM system and the effective maintenance of
10686 configuration items may be established by means of interviews with selected development staff. In
10687 conducting such interviews, the evaluator aims to gain a deeper understanding of how the CM
10688 system is used in practise as well as to confirm that the CM procedures are being applied as
10689 described in the CM documentation. Note that such interviews should complement rather than
10690 replace the examination of documentary evidence, and may not be necessary if the documentary
10691 evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is
10692 possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and
10693 records alone. This is one case where clarification may be necessary through interviews.

10694 It is expected that the evaluator will visit the development site in support of this activity.

10695 For guidance on site visits see A.4, Site Visits.

10696 **13.2.4 Evaluation of sub-activity (ALC_CMC.4)**

10697 **13.2.4.1 Objectives**

10698 The objectives of this sub-activity are to determine whether the developer has clearly identified the
 10699 TOE and its associated configuration items, and whether the ability to modify these items is
 10700 properly controlled by automated tools, thus making the CM system less susceptible to human
 10701 error or negligence.

10702 **13.2.4.2 Input**

10703 The evaluation evidence for this sub-activity is:

- 10704 a) the ST;
- 10705 b) the TOE suitable for testing;
- 10706 c) the configuration management documentation.

10707 **13.2.4.3 Action ALC_CMC.4.1E**

10708 ISO/IEC 15408-3 ALC_CMC.4.1C: *The TOE shall be labelled with its unique reference.*

10709 **13.2.4.3.1 Work unit ALC_CMC.4-1**

10710 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

10711 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST.
 10712 This could be achieved through labelled packaging or media, or by a label displayed by the
 10713 operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g.
 10714 at the point of purchase or use).

10715 The TOE may provide a method by which it can be easily identified. For example, a software TOE
 10716 may display its name and version number during the start up routine, or in response to a command
 10717 line entry. A hardware or firmware TOE may be identified by a part number physically stamped on
 10718 the TOE.

10719 Alternatively, the unique reference provided for the TOE may be the combination of the unique
 10720 reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

10721 **13.2.4.3.2 Work unit ALC_CMC.4-2**

10722 The evaluator ***shall check*** that the TOE references used are consistent.

10723 If the TOE is labelled more than once then the labels have to be consistent. For example, it should
 10724 be possible to relate any labelled guidance documentation supplied as part of the TOE to the
 10725 evaluated operational TOE. This ensures that consumers can be confident that they have purchased
 10726 the evaluated version of the TOE, that they have installed this version, and that they have the
 10727 correct version of the guidance to operate the TOE in accordance with its ST.

10728 The evaluator also verifies that the TOE reference is consistent with the ST.

10729 If this work unit is applied to a composed TOE, the following will apply. The composed TOE will not
 10730 be labelled with its unique (composite) reference, but only the individual components will be
 10731 labelled with their appropriate TOE reference. It would require further development for the
 10732 composed TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If
 10733 the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered
 10734 will not contain the composite reference. However, the composed TOE ST will include the unique

10735 reference for the composed TOE and will identify the components comprising the composed TOE
10736 through which the consumers will be able to determine whether they have the appropriate items.

10737 ISO/IEC 15408-3 ALC_CMC.4.2C: *The CM documentation shall describe the method used to uniquely*
10738 *identify the configuration items.*

10739 **13.2.4.3.3 Work unit ALC_CMC.4-3**

10740 The evaluator **shall examine** the method of identifying configuration items to determine that it
10741 describes how configuration items are uniquely identified.

10742 Procedures should describe how the status of each configuration item can be tracked throughout
10743 the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM
10744 documentation. The information included should describe:

10745 a) the method how each configuration item is uniquely identified, such that it is possible to
10746 track versions of the same configuration item;

10747 b) the method how configuration items are assigned unique identifiers and how they are
10748 entered into the CM system;

10749 c) the method to be used to identify superseded versions of a configuration item.

10750 ISO/IEC 15408-3 ALC_CMC.4.3C: *The CM system shall uniquely identify all configuration items.*

10751 **13.2.4.3.4 Work unit ALC_CMC.4-4**

10752 The evaluator **shall examine** the configuration items to determine that they are identified in a way
10753 that is consistent with the CM documentation.

10754 Assurance that the CM system uniquely identifies all configuration items is gained by examining
10755 the identifiers for the configuration items. For configuration items identified under ALC_CMS, the
10756 evaluator confirms that each configuration item possesses a unique identifier in a manner
10757 consistent with the unique identification method that is described in the CM documentation.

10758 ISO/IEC 15408-3 ALC_CMC.4.4C: *The CM system shall provide automated measures such that only*
10759 *authorised changes are made to the configuration items.*

10760 **13.2.4.3.5 Work unit ALC_CMC.4-5**

10761 The evaluator **shall examine** the CM access control measures described in the CM plan (cf.
10762 **ALC_CMC.4.6C**) to determine that they are automated and effective in preventing unauthorised
10763 access to the configuration items.

10764 The evaluator may use a number of methods to determine that the CM access control measures are
10765 effective. For example, the evaluator may exercise the access control measures to ensure that the
10766 procedures could not be bypassed. The evaluator may use the outputs generated by the CM system
10767 procedures required by **ALC_CMC.4.10C**. The evaluator may also witness a demonstration of the
10768 CM system to ensure that the access control measures employed are operating effectively.

10769 ISO/IEC 15408-3 ALC_CMC.4.5C: *The CM system shall support the production of the TOE by*
10770 *automated means.*

10771 **13.2.4.3.6 Work unit ALC_CMC.4-6**

10772 The evaluator **shall check** the CM plan (cf. **ALC_CMC.4.6C**) for automated procedures for
10773 supporting the production of the TOE.

- 10774 The term “production” applies to those processes adopted by the developer to progress the TOE
10775 from the implementation representation to a state acceptable for delivery to the end customer.
- 10776 The evaluator verifies the existence of automated production support procedures within the CM
10777 plan.
- 10778 The following are examples for automated means supporting the production of the TOE:
- 10779 — a “make” tool (as provided with many software development tools) in the case of a software
10780 TOE;
- 10781 — a tool ensuring automatically (for example by means of bar codes) that only parts are
10782 combined which indeed belong together in the case of a hardware TOE.
- 10783 **13.2.4.3.7 Work unit ALC_CMC.4-7**
- 10784 The evaluator ***shall examine*** the TOE production support procedures to determine that they are
10785 effective in ensuring that a TOE is generated that reflects its implementation representation.
- 10786 The production support procedures should describe which tools have to be used to produce the
10787 final TOE from the implementation representation in a clearly defined way. The conventions,
10788 directives, or other necessary constructs are described under ALC_TAT.
- 10789 The evaluator determines that by following the production support procedures the correct
10790 configuration items would be used to generate the TOE. For example, in a software TOE this may
10791 include checking that the automated production procedures ensure that all source files and related
10792 libraries are included in the compiled object code. Moreover, the procedures should ensure that
10793 compiler options and comparable other options are defined uniquely. For a hardware TOE, this
10794 work unit may include checking that the automatic production procedures ensure that the
10795 belonging parts are built together and no parts are missing.
- 10796 The customer can then be confident that the version of the TOE delivered for installation is derived
10797 from the implementation representation in an unambiguous way and implements the SFRs as
10798 described in the ST.
- 10799 The evaluator should bear in mind that the CM system need not necessarily possess the capability
10800 to produce the TOE, but should provide support for the process that will help reduce the
10801 probability of human error.
- 10802 ISO/IEC 15408-3 ALC_CMC.4.6C: *The CM documentation shall include a CM plan.*
- 10803 **13.2.4.3.8 Work unit ALC_CMC.4-8**
- 10804 The evaluator ***shall check*** that the CM documentation provided includes a CM plan.
- 10805 The CM plan does not need to be contained within a single document, but it is recommended that
10806 there is a separate document that describes where the various parts of the CM plan can be found. If
10807 the CM plan is provided by a set of documents, the list in the following work unit gives guidance
10808 regarding the required content.
- 10809 ISO/IEC 15408-3 ALC_CMC.4.7C: *The CM plan shall describe how the CM system is used for the*
10810 *development of the TOE.*
- 10811 **13.2.4.3.9 Work unit ALC_CMC.4-9**
- 10812 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used
10813 for the development of the TOE.

- 10814 The descriptions contained in a CM plan include, if applicable:
- 10815 a) all activities performed in the TOE development that are subject to configuration
10816 management procedures (e.g. creation, modification or deletion of a configuration item,
10817 data-backup, archiving);
 - 10818 b) which means (e.g. CM tools, forms) have to be made available;
 - 10819 c) the usage of the CM tools: the necessary details for a user of the CM system to be able to
10820 operate the CM tools correctly in order to maintain the integrity of the TOE;
 - 10821 d) the production support procedures;
 - 10822 e) which other objects (development components, tools, assessment environments, etc) are
10823 taken under CM control;
 - 10824 f) the roles and responsibilities of individuals required to perform operations on individual
10825 configuration items (different roles may be identified for different types of configuration
10826 items (e.g. design documentation or source code));
 - 10827 g) how CM instances (e.g. change control boards, interface control working groups) are
10828 introduced and staffed;
 - 10829 h) the description of the change management;
 - 10830 i) the procedures that are used to ensure that only authorised individuals can make changes
10831 to configuration items;
 - 10832 j) the procedures that are used to ensure that concurrency problems do not occur as a
10833 result of simultaneous changes to configuration items;
 - 10834 k) the evidence that is generated as a result of application of the procedures. For example,
10835 for a change to a configuration item, the CM system might record a description of the
10836 change, accountability for the change, identification of all configuration items affected,
10837 status (e.g. pending or completed), and date and time of the change. This might be
10838 recorded in an audit trail of changes made or change control records;
 - 10839 l) the approach to version control and unique referencing of TOE versions (e.g. covering the
10840 release of patches in operating systems, and the subsequent detection of their
10841 application).
- 10842 ISO/IEC 15408-3 ALC_CMC.4.8C: *The CM plan shall describe the procedures used to accept modified*
10843 *or newly created configuration items as part of the TOE.*
- 10844 **13.2.4.3.10 Work unit ALC_CMC.4-10**
- 10845 The evaluator ***shall examine*** the CM plan to determine that it describes the procedures used to
10846 accept modified or newly created configuration items as parts of the TOE.
- 10847 The descriptions of the acceptance procedures in the CM plan should include the developer roles or
10848 individuals responsible for the acceptance and the criteria to be used for acceptance. They should
10849 take into account all acceptance situations that may occur, in particular:
- 10850 a) accepting an item into the CM system for the first time, in particular inclusion of software,
10851 firmware and hardware components from other manufacturers into the TOE
10852 ("integration");

- 10853 b) moving configuration items to the next life-cycle phase at each stage of the construction of
10854 the TOE (e.g. module, subsystem, system);
- 10855 c) subsequent to transports between different development sites.
- 10856 If this work unit is applied to a dependent component that is going to be integrated in a composed
10857 TOE, the CM plan should consider the control of base components obtained by the dependent TOE
10858 developer.
- 10859 When obtaining the components the evaluators are to verify the following:
- 10860 a) Transfer of each base component from the base component developer to the integrator
10861 (dependent TOE developer) was performed in accordance with the base component
10862 TOE's secure delivery procedures, as reported in the base component TOE certification
10863 report.
- 10864 b) The component received has the same identifiers as those stated in the ST and
10865 Certification Report for the component TOE.
- 10866 c) All additional material required by a developer for composition (integration) is provided.
10867 This is to include the necessary extract of the component TOE's functional specification.
- 10868 ISO/IEC 15408-3 ALC_CMC.4.9C: *The evidence shall demonstrate that all configuration items are*
10869 *being maintained under the CM system.*
- 10870 **13.2.4.3.11 Work unit ALC_CMC.4-11**
- 10871 The evaluator **shall check** that the configuration items identified in the configuration list are being
10872 maintained by the CM system.
- 10873 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator
10874 should check that for each type of configuration item (e.g. design documents or source code
10875 modules) contained in the configuration list there are examples of the evidence generated by the
10876 procedures described in the CM plan. In this case, the approach to sampling will depend upon the
10877 level of granularity used in the CM system to control CM items. Where, for example, 10,000 source
10878 code modules are identified in the configuration list, a different sampling strategy needs to be
10879 applied compared to the case in which there are only 5, or even 1. The emphasis of this activity
10880 should be on ensuring that the CM system is being operated correctly, rather than on the detection
10881 of any minor error.
- 10882 For guidance on sampling see A.2, Sampling.
- 10883 ISO/IEC 15408-3 ALC_CMC.4.10C: *The evidence shall demonstrate that the CM system is being*
10884 *operated in accordance with the CM plan.*
- 10885 **13.2.4.3.12 Work unit ALC_CMC.4-12**
- 10886 The evaluator **shall check** the CM documentation to ascertain that it includes the CM system
10887 records identified by the CM plan.
- 10888 The output produced by the CM system should provide the evidence that the evaluator needs to be
10889 confident that the CM plan is being applied, and also that all configuration items are being
10890 maintained by the CM system as required by **ALC_CMC.4.9C**. Example output could include change
10891 control forms, or configuration item access approval forms.

10892 **13.2.4.3.13 Work unit ALC_CMC.4-13**

10893 The evaluator **shall examine** the evidence to determine that the CM system is being operated in
10894 accordance with the CM plan.

10895 The evaluator should select and examine a sample of evidence covering each type of CM-relevant
10896 operation that has been performed on a configuration item (e.g. creation, modification, deletion,
10897 reversion to an earlier version) to confirm that all operations of the CM system have been carried
10898 out in line with documented procedures. The evaluator confirms that the evidence includes all the
10899 information identified for that operation in the CM plan. Examination of the evidence may require
10900 access to a CM tool that is used. The evaluator may choose to sample the evidence.

10901 For guidance on sampling see A.2, Sampling.

10902 Further confidence in the correct operation of the CM system and the effective maintenance of
10903 configuration items may be established by means of interviews with selected development staff. In
10904 conducting such interviews, the evaluator aims to gain a deeper understanding of how the CM
10905 system is used in practise as well as to confirm that the CM procedures are being applied as
10906 described in the CM documentation. Note that such interviews should complement rather than
10907 replace the examination of documentary evidence, and may not be necessary if the documentary
10908 evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is
10909 possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and
10910 records alone. This is one case where clarification may be necessary through interviews.

10911 It is expected that the evaluator will visit the development site in support of this activity.

10912 For guidance on site visits see A.4, Site Visits.

10913 **13.2.5 Evaluation of sub-activity (ALC_CMC.5)**

10914 **13.2.5.1 Objectives**

10915 The objectives of this sub-activity are to determine whether the developer has clearly identified the
10916 TOE and its associated configuration items, and whether the ability to modify these items is
10917 properly controlled by automated tools, thus making the CM system less susceptible to human
10918 error or negligence.

10919 **13.2.5.2 Input**

10920 The evaluation evidence for this sub-activity is:

- 10921 a) the ST;
- 10922 b) the TOE suitable for testing;
- 10923 c) the configuration management documentation.

10924 **13.2.5.3 Action ALC_CMC.5.1E**

10925 ISO/IEC 15408-3 ALC_CMC.5.1C: *The TOE shall be labelled with its unique reference.*

10926 **13.2.5.3.1 Work unit ALC_CMC.5-1**

10927 The evaluator **shall check** that the TOE provided for evaluation is labelled with its reference.

10928 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST.
10929 This could be achieved through labelled packaging or media, or by a label displayed by the

- 10930 operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g.
10931 at the point of purchase or use).
- 10932 The TOE may provide a method by which it can be easily identified. For example, a software TOE
10933 may display its name and version number during the start up routine, or in response to a command
10934 line entry. A hardware or firmware TOE may be identified by a part number physically stamped on
10935 the TOE.
- 10936 Alternatively, the unique reference provided for the TOE may be the combination of the unique
10937 reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).
- 10938 **13.2.5.3.2 Work unit ALC_CMC.5-2**
- 10939 The evaluator ***shall check*** that the TOE references used are consistent.
- 10940 If the TOE is labelled more than once then the labels have to be consistent. For example, it should
10941 be possible to relate any labelled guidance documentation supplied as part of the TOE to the
10942 evaluated operational TOE. This ensures that consumers can be confident that they have purchased
10943 the evaluated version of the TOE, that they have installed this version, and that they have the
10944 correct version of the guidance to operate the TOE in accordance with its ST.
- 10945 The evaluator also verifies that the TOE reference is consistent with the ST.
- 10946 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will
10947 not be labelled with its unique (composite) reference, but only the individual components will be
10948 labelled with their appropriate TOE reference. It would require further development for the IT TOE
10949 to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed
10950 TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain
10951 the composite reference. However, the composed TOE ST will include the unique reference for the
10952 composed TOE and will identify the components comprising the composed TOE through which the
10953 consumers will be able to determine whether they have the appropriate items.
- 10954 ISO/IEC 15408-3 ALC_CMC.5.2C: *The CM documentation shall describe the method used to uniquely*
10955 *identify the configuration items.*
- 10956 **13.2.5.3.3 Work unit ALC_CMC.5-3**
- 10957 The evaluator ***shall examine*** the method of identifying configuration items to determine that it
10958 describes how configuration items are uniquely identified.
- 10959 Procedures should describe how the status of each configuration item can be tracked throughout
10960 the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM
10961 documentation. The information included should describe:
- 10962 a) the method how each configuration item is uniquely identified, such that it is possible to
10963 track versions of the same configuration item;
- 10964 b) the method how configuration items are assigned unique identifiers and how they are
10965 entered into the CM system;
- 10966 c) the method to be used to identify superseded versions of a configuration item.
- 10967 ISO/IEC 15408-3 ALC_CMC.5.3C: *The CM documentation shall justify that the acceptance procedures*
10968 *provide for an adequate and appropriate review of changes to all configuration items.*

10969 **13.2.5.3.4 Work unit ALC_CMC.5-4**

10970 The evaluator **shall examine** the CM documentation to determine that it justifies that the
10971 acceptance procedures provide for an adequate and appropriate review of changes to all
10972 configuration items.

10973 The CM documentation should make it sufficiently clear that by following the acceptance
10974 procedures only parts of adequate quality are incorporated into the TOE.

10975 ISO/IEC 15408-3 ALC_CMC.5.4C: *The CM system shall uniquely identify all configuration items.*

10976 **13.2.5.3.5 Work unit ALC_CMC.5-5**

10977 The evaluator **shall examine** the configuration items to determine that they are identified in a way
10978 that is consistent with the CM documentation.

10979 Assurance that the CM system uniquely identifies all configuration items is gained by examining
10980 the identifiers for the configuration items. For both configuration items that comprise the TOE, and
10981 drafts of configuration items that are submitted by the developer as evaluation evidence, the
10982 evaluator confirms that each configuration item possesses a unique identifier in a manner
10983 consistent with the unique identification method that is described in the CM documentation.

10984 ISO/IEC 15408-3 ALC_CMC.5.5C: *The CM system shall provide automated measures such that only*
10985 *authorised changes are made to the configuration items.*

10986 **13.2.5.3.6 Work unit ALC_CMC.5-6**

10987 The evaluator **shall examine** the CM access control measures described in the CM plan (cf.
10988 **ALC_CMC.5.12C**) to determine that they are automated and effective in preventing unauthorised
10989 access to the configuration items.

10990 The evaluator may use a number of methods to determine that the CM access control measures are
10991 effective. For example, the evaluator may exercise the access control measures to ensure that the
10992 procedures could not be bypassed. The evaluator may use the outputs generated by the CM system
10993 procedures required by **ALC_CMC.5.16C**. The evaluator may also witness a demonstration of the
10994 CM system to ensure that the access control measures employed are operating effectively.

10995 ISO/IEC 15408-3 ALC_CMC.5.6C: *The CM system shall support the production of the TOE by*
10996 *automated means.*

10997 **13.2.5.3.7 Work unit ALC_CMC.5-7**

10998 The evaluator **shall check** the CM plan (cf. **ALC_CMC.5.12C**) for automated procedures for
10999 supporting the production of the TOE.

11000 The term “production” applies to those processes adopted by the developer to progress the TOE
11001 from the implementation representation to a state acceptable for delivery to the end customer.

11002 The evaluator verifies the existence of automated production support procedures within the CM
11003 plan.

11004 The following are examples for automated means supporting the production of the TOE:

11005 — a “make” tool (as provided with many software development tools) in the case of a software
11006 TOE;

11007 — a tool ensuring automatically (for example by means of bar codes) that only parts are
11008 combined which indeed belong together in the case of a hardware TOE.

11009 **13.2.5.3.8 Work unit ALC_CMC.5-8**

11010 The evaluator **shall examine** the TOE production support procedures to determine that they are
11011 effective in ensuring that a TOE is generated that reflects its implementation representation.

11012 The production support procedures should describe which tools have to be used to produce the
11013 final TOE from the implementation representation in a clearly defined way. The conventions,
11014 directives, or other necessary constructs are described under ALC_TAT.

11015 The evaluator determines that by following the production support procedures the correct
11016 configuration items would be used to generate the TOE. For example, in a software TOE this may
11017 include checking that the automated production procedures ensure that all source files and related
11018 libraries are included in the compiled object code. Moreover, the procedures should ensure that
11019 compiler options and comparable other options are defined uniquely. For a hardware TOE, this
11020 work unit may include checking that the automatic production procedures ensure that the
11021 belonging parts are built together and no parts are missing.

11022 The customer can then be confident that the version of the TOE delivered for installation is derived
11023 from the implementation representation in an unambiguous way and implements the SFRs as
11024 described in the ST.

11025 The evaluator should bear in mind that the CM system need not necessarily possess the capability
11026 to produce the TOE, but should provide support for the process that will help reduce the
11027 probability of human error.

11028 ISO/IEC 15408-3 ALC_CMC.5.7C: *The CM system shall ensure that the person responsible for*
11029 *accepting a configuration item into CM is not the person who developed it.*

11030 **13.2.5.3.9 Work unit ALC_CMC.5-9**

11031 The evaluator **shall examine** the CM system to determine that it ensures that the person
11032 responsible for accepting a configuration item is not the person who developed it.

11033 The acceptance procedures describe who is responsible for accepting a configuration item. From
11034 these descriptions, the evaluator should be able to determine that the person who developed a
11035 configuration item is in no case responsible for its acceptance.

11036 ISO/IEC 15408-3 ALC_CMC.5.8C: *The CM system shall identify the configuration items that comprise*
11037 *the TSF.*

11038 **13.2.5.3.10 Work unit ALC_CMC.5-10**

11039 The evaluator **shall examine** the CM system to determine that it identifies the configuration items
11040 that comprise the TSF.

11041 The CM documentation should describe how the CM system identifies the configuration items that
11042 comprise the TSF. The evaluator should select a sample of configuration items covering each type
11043 of items, particularly containing TSF and non-TSF items, and check that they are correctly classified
11044 by the CM system.

11045 For guidance on sampling see A.2, Sampling.

11046 ISO/IEC 15408-3 ALC_CMC.5.9C: *The CM system shall support the audit of all changes to the TOE by*
11047 *automated means, including the originator, date, and time in the audit trail.*

11048 **13.2.5.3.11 Work unit ALC_CMC.5-11**

11049 The evaluator **shall examine** the CM system to determine that it supports the audit of all changes
11050 to the TOE by automated means, including the originator, date, and time in the audit trail.

11051 The evaluator should inspect a sample of audit trails and check, if they contain the minimum
11052 information.

11053 ISO/IEC 15408-3 ALC_CMC.5.10C: *The CM system shall provide an automated means to identify all*
11054 *other configuration items that are affected by the change of a given configuration item.*

11055 **13.2.5.3.12 Work unit ALC_CMC.5-12**

11056 The evaluator **shall examine** the CM system to determine that it provides an automated means to
11057 identify all other configuration items that are affected by the change of a given configuration item.

11058 The CM documentation should describe how the CM system identifies all other configuration items
11059 that are affected by the change of a given configuration item. The evaluator should select a sample
11060 of configuration items, covering all types of items, and exercise the automated means to determine
11061 that it identifies all items that are affected by the change of the selected item.

11062 For guidance on sampling see A.2, Sampling.

11063 ISO/IEC 15408-3 ALC_CMC.5.11C: *The CM system shall be able to identify the version of the*
11064 *implementation representation from which the TOE is generated.*

11065 **13.2.5.3.13 Work unit ALC_CMC.5-13**

11066 The evaluator **shall examine** the CM system to determine that it is able to identify the version of
11067 the implementation representation from which the TOE is generated.

11068 The CM documentation should describe how the CM system identifies the version of the
11069 implementation representation from which the TOE is generated. The evaluator should select a
11070 sample of the parts used to produce the TOE and should apply the CM system to verify that it
11071 identifies the corresponding implementation representation in the correct version.

11072 For guidance on sampling see A.2, Sampling.

11073 ISO/IEC 15408-3 ALC_CMC.5.12C: *The CM documentation shall include a CM plan.*

11074 **13.2.5.3.14 Work unit ALC_CMC.5-14**

11075 The evaluator **shall check** that the CM documentation provided includes a CM plan.

11076 The CM plan needs not to be a connected document, but it is recommended that there is a single
11077 document that describes where the various parts of the CM plan can be found. If the CM plan is no
11078 single document, the list in the following work unit gives hints regarding which context is expected.

11079 ISO/IEC 15408-3 ALC_CMC.5.13C: *The CM plan shall describe how the CM system is used for the*
11080 *development of the TOE.*

11081 **13.2.5.3.15 Work unit ALC_CMC.5-15**

11082 The evaluator **shall examine** the CM plan to determine that it describes how the CM system is used
11083 for the development of the TOE.

11084 The descriptions contained in a CM plan include, if applicable:

- 11085 a) all activities performed in the TOE development that are subject to configuration
11086 management procedures (e.g. creation, modification or deletion of a configuration item,
11087 data-backup, archiving);
- 11088 b) which means (e.g. CM tools, forms) have to be made available;
- 11089 c) the usage of the CM tools: the necessary details for a user of the CM system to be able to
11090 operate the CM tools correctly in order to maintain the integrity of the TOE;
- 11091 d) the production support procedures;
- 11092 e) which other objects (development components, tools, assessment environments, etc) are
11093 taken under CM control;
- 11094 f) the roles and responsibilities of individuals required to perform operations on individual
11095 configuration items (different roles may be identified for different types of configuration
11096 items (e.g. design documentation or source code));
- 11097 g) how CM instances (e.g. change control boards, interface control working groups) are
11098 introduced and staffed;
- 11099 h) the description of the change management;
- 11100 i) the procedures that are used to ensure that only authorised individuals can make changes
11101 to configuration items;
- 11102 j) the procedures that are used to ensure that concurrency problems do not occur as a
11103 result of simultaneous changes to configuration items;
- 11104 k) the evidence that is generated as a result of application of the procedures. For example,
11105 for a change to a configuration item, the CM system might record a description of the
11106 change, accountability for the change, identification of all configuration items affected,
11107 status (e.g. pending or completed), and date and time of the change. This might be
11108 recorded in an audit trail of changes made or change control records;
- 11109 l) the approach to version control and unique referencing of TOE versions (e.g. covering the
11110 release of patches in operating systems, and the subsequent detection of their
11111 application).
- 11112 ISO/IEC 15408-3 ALC_CMC.5.14C: *The CM plan shall describe the procedures used to accept modified*
11113 *or newly created configuration items as part of the TOE.*
- 11114 **13.2.5.3.16 Work unit ALC_CMC.5-16**
- 11115 The evaluator ***shall examine*** the CM plan to determine that it describes the procedures used to
11116 accept modified or newly created configuration items as parts of the TOE.
- 11117 The descriptions of the acceptance procedures in the CM plan should include the developer roles or
11118 individuals responsible for the acceptance and the criteria to be used for acceptance. They should
11119 take into account all acceptance situations that may occur, in particular:
- 11120 a) accepting an item into the CM system for the first time, in particular inclusion of software,
11121 firmware and hardware components from other manufacturers into the TOE
11122 ("integration");
- 11123 b) moving configuration items to the next life-cycle phase at each stage of the construction of
11124 the TOE (e.g. module, subsystem, system);

- 11125 c) subsequent to transports between different development sites.
- 11126 ISO/IEC 15408-3 ALC_CMC.5.15C: *The evidence shall demonstrate that all configuration items are*
11127 *being maintained under the CM system.*
- 11128 **13.2.5.3.17 Work unit ALC_CMC.5-17**
- 11129 The evaluator **shall check** that the configuration items identified in the configuration list are being
11130 maintained by the CM system.
- 11131 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator
11132 should check that for each type of configuration item (e.g. design documents or source code
11133 modules) contained in the configuration list there are examples of the evidence generated by the
11134 procedures described in the CM plan. In this case, the approach to sampling will depend upon the
11135 level of granularity used in the CM system to control CM items. Where, for example, 10,000 source
11136 code modules are identified in the configuration list, a different sampling strategy needs to be
11137 applied compared to the case in which there are only 5, or even 1. The emphasis of this activity
11138 should be on ensuring that the CM system is being operated correctly, rather than on the detection
11139 of any minor error.
- 11140 For guidance on sampling see A.2, Sampling.
- 11141 ISO/IEC 15408-3 ALC_CMC.5.16C: *The evidence shall demonstrate that the CM system is being*
11142 *operated in accordance with the CM plan.*
- 11143 **13.2.5.3.18 Work unit ALC_CMC.5-18**
- 11144 The evaluator **shall check** the CM documentation to ascertain that it includes the CM system
11145 records identified by the CM plan.
- 11146 The output produced by the CM system should provide the evidence that the evaluator needs to be
11147 confident that the CM plan is being applied, and also that all configuration items are being
11148 maintained by the CM system as required by **ALC_CMC.5.15C**. Example output could include change
11149 control forms, or configuration item access approval forms.
- 11150 **13.2.5.3.19 Work unit ALC_CMC.5-19**
- 11151 The evaluator **shall examine** the evidence to determine that the CM system is being operated in
11152 accordance with the CM plan.
- 11153 The evaluator should select and examine a sample of evidence covering each type of CM-relevant
11154 operation that has been performed on a configuration item (e.g. creation, modification, deletion,
11155 reversion to an earlier version) to confirm that all operations of the CM system have been carried
11156 out in line with documented procedures. The evaluator confirms that the evidence includes all the
11157 information identified for that operation in the CM plan. Examination of the evidence may require
11158 access to a CM tool that is used. The evaluator may choose to sample the evidence.
- 11159 For guidance on sampling see A.2, Sampling.
- 11160 Further confidence in the correct operation of the CM system and the effective maintenance of
11161 configuration items may be established by means of interviews with selected development staff. In
11162 conducting such interviews, the evaluator aims to gain a deeper understanding of how the CM
11163 system is used in practise as well as to confirm that the CM procedures are being applied as
11164 described in the CM documentation. Note that such interviews should complement rather than
11165 replace the examination of documentary evidence, and may not be necessary if the documentary
11166 evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is
11167 possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and
11168 records alone. This is one case where clarification may be necessary through interviews.

- 11169 It is expected that the evaluator will visit the development site in support of this activity.
- 11170 For guidance on site visits see A.4, Site Visits.
- 11171 **13.2.5.4 Action ALC_CMC.5.2E**
- 11172 **13.2.5.4.1 Work unit ALC_CMC.5-20**
- 11173 The evaluator **shall examine** the production support procedures to determine that by following
11174 these procedures a TOE would be produced like that one provided by the developer for testing
11175 activities.
- 11176 If the TOE is a small software TOE and production consists of compiling and linking, the evaluator
11177 might confirm the adequacy of the production support procedures by reapplying them himself.
- 11178 If the production process of the TOE is more complicated (as for example in the case of a smart
11179 card), but has already started, the evaluator should inspect the application of the production
11180 support procedures during a visit of the development site. They might compare a copy of the TOE
11181 produced in their presence with the samples used for their testing activities.
- 11182 For guidance on site visits see A.4, Site Visits.
- 11183 Otherwise the evaluator's determination should be based on the documentary evidence provided
11184 by the developer.
- 11185 This work unit may be performed in conjunction with the evaluation activities under
11186 Implementation representation (ADV_IMP).
- 11187 **13.3 CM scope (ALC_CMS)**
- 11188 **13.3.1 Evaluation of sub-activity (ALC_CMS.1)**
- 11189 **13.3.1.1 Objectives**
- 11190 The objective of this sub-activity is to determine whether the developer performs configuration
11191 management on the TOE and the evaluation evidence. These configuration items are controlled in
11192 accordance with CM capabilities (ALC_CMC).
- 11193 **13.3.1.2 Input**
- 11194 The evaluation evidence for this sub-activity is:
- 11195 a) the ST;
- 11196 b) the configuration list.
- 11197 **13.3.1.3 Action ALC_CMS.1.1E**
- 11198 ISO/IEC 15408-3 ALC_CMS.1.1C: *The configuration list shall include the following: the TOE itself; and*
11199 *the evaluation evidence required by the SARs.*
- 11200 **13.3.1.3.1 Work unit ALC_CMS.1-1**
- 11201 The evaluator **shall check** that the configuration list includes the following set of items:
- 11202 a) the TOE itself;
- 11203 b) the evaluation evidence required by the SARs in the ST.

11204 ISO/IEC 15408-3 ALC_CMS.1.2C: *The configuration list shall uniquely identify the configuration items.*

11205 **13.3.1.3.2 Work unit ALC_CMS.1-2**

11206 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each
11207 configuration item.

11208 The configuration list contains sufficient information to uniquely identify which version of each
11209 item has been used (typically a version number). Use of this list will enable the evaluator to check
11210 that the correct configuration items, and the correct version of each item, have been used during
11211 the evaluation.

11212 **13.3.2 Evaluation of sub-activity (ALC_CMS.2)**

11213 **13.3.2.1 Objectives**

11214 The objective of this sub-activity is to determine whether the configuration list includes the TOE,
11215 the parts that comprise the TOE, and the evaluation evidence. These configuration items are
11216 controlled in accordance with CM capabilities (ALC_CMC).

11217 **13.3.2.2 Input**

11218 The evaluation evidence for this sub-activity is:

11219 a) the ST;

11220 b) the configuration list.

11221 **13.3.2.3 Action ALC_CMS.2.1E**

11222 ISO/IEC 15408-3 ALC_CMS.2.1C: *The configuration list shall include the following: the TOE itself; the*
11223 *evaluation evidence required by the SARs; and the parts that comprise the TOE.*

11224 **13.3.2.3.1 Work unit ALC_CMS.2-1**

11225 The evaluator ***shall check*** that the configuration list includes the following set of items:

11226 a) the TOE itself;

11227 b) the parts that comprise the TOE;

11228 c) the evaluation evidence required by the SARs.

11229 ISO/IEC 15408-3 ALC_CMS.2.2C: *The configuration list shall uniquely identify the configuration items.*

11230 **13.3.2.3.2 Work unit ALC_CMS.2-2**

11231 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each
11232 configuration item.

11233 The configuration list contains sufficient information to uniquely identify which version of each
11234 item has been used (typically a version number). Use of this list will enable the evaluator to check
11235 that the correct configuration items, and the correct version of each item, have been used during
11236 the evaluation.

11237 ISO/IEC 15408-3 ALC_CMS.2.3C: *For each TSF relevant configuration item, the configuration list*
11238 *shall indicate the developer of the item.*

11239 **13.3.2.3.3 Work unit ALC_CMS.2-3**

11240 The evaluator **shall check** that the configuration list indicates the developer of each TSF relevant
11241 configuration item.

11242 If only one developer is involved in the development of the TOE, this work unit is not applicable,
11243 and is therefore considered to be satisfied.

11244 **13.3.3 Evaluation of sub-activity (ALC_CMS.3)**11245 **13.3.3.1 Objectives**

11246 The objective of this sub-activity is to determine whether the configuration list includes the TOE,
11247 the parts that comprise the TOE, the TOE implementation representation, and the evaluation
11248 evidence. These configuration items are controlled in accordance with CM capabilities (ALC_CMC).

11249 **13.3.3.2 Input**

11250 The evaluation evidence for this sub-activity is:

11251 a) the ST;

11252 b) the configuration list.

11253 **13.3.3.3 Action ALC_CMS.3.1E**

11254 ISO/IEC 15408-3 ALC_CMS.3.1C: *The configuration list shall include the following: the TOE itself; the*
11255 *evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation*
11256 *representation.*

11257 **13.3.3.3.1 Work unit ALC_CMS.3-1**

11258 The evaluator **shall check** that the configuration list includes the following set of items:

11259 a) the TOE itself;

11260 b) the parts that comprise the TOE;

11261 c) the TOE implementation representation;

11262 d) the evaluation evidence required by the SARs in the ST.

11263 ISO/IEC 15408-3 ALC_CMS.3.2C: *The configuration list shall uniquely identify the configuration items.*

11264 **13.3.3.3.2 Work unit ALC_CMS.3-2**

11265 The evaluator **shall examine** the configuration list to determine that it uniquely identifies each
11266 configuration item.

11267 The configuration list contains sufficient information to uniquely identify which version of each
11268 item has been used (typically a version number). Use of this list will enable the evaluator to check
11269 that the correct configuration items, and the correct version of each item, have been used during
11270 the evaluation.

11271 ISO/IEC 15408-3 ALC_CMS.3.3C: *For each TSF relevant configuration item, the configuration list*
11272 *shall indicate the developer of the item.*

11273 **13.3.3.3.3 Work unit ALC_CMS.3-3**

11274 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant
11275 configuration item.

11276 If only one developer is involved in the development of the TOE, this work unit is not applicable,
11277 and is therefore considered to be satisfied.

11278 **13.3.4 Evaluation of sub-activity (ALC_CMS.4)**

11279 **13.3.4.1 Objectives**

11280 The objective of this sub-activity is to determine whether the configuration list includes the TOE,
11281 the parts that comprise the TOE, the TOE implementation representation, security flaws, and the
11282 evaluation evidence. These configuration items are controlled in accordance with CM capabilities
11283 (ALC_CMC).

11284 **13.3.4.2 Input**

11285 The evaluation evidence for this sub-activity is:

- 11286 a) the ST;
- 11287 b) the configuration list.

11288 **13.3.4.3 Action ALC_CMS.4.1E**

11289 ISO/IEC 15408-3 ALC_CMS.4.1C: *The configuration list shall include the following: the TOE itself; the*
11290 *evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation*
11291 *representation; and security flaw reports and resolution status.*

11292 **13.3.4.3.1 Work unit ALC_CMS.4-1**

11293 The evaluator ***shall check*** that the configuration list includes the following set of items:

- 11294 a) the TOE itself;
- 11295 b) the parts that comprise the TOE;
- 11296 c) the TOE implementation representation;
- 11297 d) the evaluation evidence required by the SARs in the ST;
- 11298 e) the documentation used to record details of reported security flaws associated with the
11299 implementation (e.g., problem status reports derived from a developer's problem
11300 database).

11301 ISO/IEC 15408-3 ALC_CMS.4.2C: *The configuration list shall uniquely identify the configuration items.*

11302 **13.3.4.3.2 Work unit ALC_CMS.4-2**

11303 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each
11304 configuration item.

11305 The configuration list contains sufficient information to uniquely identify which version of each
11306 item has been used (typically a version number). Use of this list will enable the evaluator to check
11307 that the correct configuration items, and the correct version of each item, have been used during
11308 the evaluation.

11309 ISO/IEC 15408-3 ALC_CMS.4.3C: *For each TSF relevant configuration item, the configuration list*
 11310 *shall indicate the developer of the item.*

11311 **13.3.4.3.3 Work unit ALC_CMS.4-3**

11312 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant
 11313 configuration item.

11314 If only one developer is involved in the development of the TOE, this work unit is not applicable,
 11315 and is therefore considered to be satisfied.

11316 **13.3.5 Evaluation of sub-activity (ALC_CMS.5)**

11317 **13.3.5.1 Objectives**

11318 The objective of this sub-activity is to determine whether the configuration list includes the TOE,
 11319 the parts that comprise the TOE, the TOE implementation representation, security flaws,
 11320 development tools and related information, and the evaluation evidence. These configuration items
 11321 are controlled in accordance with CM capabilities (ALC_CMC).

11322 **13.3.5.2 Input**

11323 The evaluation evidence for this sub-activity is:

- 11324 a) the ST;
- 11325 b) the configuration list.

11326 **13.3.5.3 Action ALC_CMS.5.1E**

11327 ISO/IEC 15408-3 ALC_CMS.5.1C: *The configuration list shall include the following: the TOE itself; the*
 11328 *evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation*
 11329 *representation; security flaw reports and resolution status; and development tools and related*
 11330 *information.*

11331 **13.3.5.3.1 Work unit ALC_CMS.5-1**

11332 The evaluator ***shall check*** that the configuration list includes the following set of items:

- 11333 a) the TOE itself;
- 11334 b) the parts that comprise the TOE;
- 11335 c) the TOE implementation representation;
- 11336 d) the evaluation evidence required by the SARs in the ST;
- 11337 e) the documentation used to record details of reported security flaws associated with the
 11338 implementation (e.g., problem status reports derived from a developer's problem
 11339 database);
- 11340 f) all tools (incl. test software, if applicable) involved in the development and production of
 11341 the TOE including the names, versions, configurations and roles of each development tool,
 11342 and related documentation.

11343 For a software TOE, “development tools” are usually programming languages and compiler and
 11344 “related documentation” comprises compiler and linker options. For a hardware TOE,

11345 “development tools” might be hardware design languages, simulation and synthesis tools,
11346 compilers, and “related documentation” might comprise compiler options again.

11347 ISO/IEC 15408-3 ALC_CMS.5.2C: *The configuration list shall uniquely identify the configuration items.*

11348 **13.3.5.3.2 Work unit ALC_CMS.5-2**

11349 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each
11350 configuration item.

11351 The configuration list contains sufficient information to uniquely identify which version of each
11352 item has been used (typically a version number). Use of this list will enable the evaluator to check
11353 that the correct configuration items, and the correct version of each item, have been used during
11354 the evaluation.

11355 ISO/IEC 15408-3 ALC_CMS.5.3C: *For each TSF relevant configuration item, the configuration list*
11356 *shall indicate the developer of the item.*

11357 **13.3.5.3.3 Work unit ALC_CMS.5-3**

11358 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant
11359 configuration item.

11360 If only one developer is involved in the development of the TOE, this work unit is not applicable,
11361 and is therefore considered to be satisfied.

11362 **13.4 Delivery (ALC_DEL)**

11363 **13.4.1 Evaluation of sub-activity (ALC_DEL.1)**

11364 **13.4.1.1 Objectives**

11365 The objective of this sub-activity is to determine whether the delivery documentation describes all
11366 procedures used to maintain security of the TOE when distributing the TOE to the user.

11367 **13.4.1.2 Input**

11368 The evaluation evidence for this sub-activity is:

11369 a) the ST;

11370 b) the delivery documentation.

11371 **13.4.1.3 Action ALC_DEL.1.1E**

11372 ISO/IEC 15408-3 ALC_DEL.1.1C: *The delivery documentation shall describe all procedures that are*
11373 *necessary to maintain security when distributing versions of the TOE to the consumer.*

11374 **13.4.1.3.1 Work unit ALC_DEL.1-1**

11375 The evaluator ***shall examine*** the delivery documentation to determine that it describes all
11376 procedures that are necessary to maintain security when distributing versions of the TOE or parts
11377 of it to the consumer.

11378 The delivery documentation describes proper procedures to maintain security of the TOE during
11379 transfer of the TOE or its component parts and to determine the identification of the TOE.

- 11380 The delivery documentation should cover the entire TOE, but may contain different procedures for
11381 different parts of the TOE. The evaluation should consider the totality of procedures.
- 11382 The delivery procedures should be applicable across all phases of delivery from the production
11383 environment to the installation environment (e.g. packaging, storage and distribution). Standard
11384 commercial practise for packaging and delivery may be acceptable. This includes shrink wrapped
11385 packaging, a security tape or a sealed envelope. For the distribution, physical (e.g. public mail or a
11386 private distribution service) or electronic (e.g. electronic mail or downloading off the Internet)
11387 procedures may be used.
- 11388 Cryptographic checksums or a software signature may be used by the developer to ensure that
11389 tampering or masquerading can be detected. Tamper proof seals additionally indicate if the
11390 confidentiality has been broken. For software TOEs, confidentiality might be assured by using
11391 encryption. If availability is of concern, a secure transportation might be required.
- 11392 Interpretation of the term “necessary to maintain security” will need to consider:
- 11393 — The nature of the TOE (e.g. whether it is software or hardware).
- 11394 — The overall security level stated for the TOE by the chosen level of the Vulnerability
11395 Assessment. If the TOE is required to be resistant against attackers of a certain potential in its
11396 intended environment, this should also apply to the delivery of the TOE. The evaluator should
11397 determine that a balanced approach has been taken, such that delivery does not present a
11398 weak point in an otherwise secure development process.
- 11399 — The security objectives provided by the ST. The emphasis in the delivery documentation is
11400 likely to be on measures related to integrity, as integrity of the TOE is always important.
11401 However, confidentiality and availability of the delivery will be of concern in the delivery of
11402 some TOEs; procedures relating to these aspects of the secure delivery should also be
11403 discussed in the procedures.
- 11404 **13.4.1.4 Implied evaluator action**
- 11405 ISO/IEC 15408-3 ALC_DEL.1.2D: *The developer shall use the delivery procedures.*
- 11406 **13.4.1.4.1 Work unit ALC_DEL.1-2**
- 11407 The evaluator ***shall examine*** aspects of the delivery process to determine that the delivery
11408 procedures are used.
- 11409 The approach taken by the evaluator to check the application of delivery procedures will depend
11410 on the nature of the TOE, and the delivery process itself. In addition to examination of the
11411 procedures themselves, the evaluator seeks some assurance that they are applied in practise. Some
11412 possible approaches are:
- 11413 a) a visit to the distribution site(s) where practical application of the procedures may be
11414 observed;
- 11415 b) examination of the TOE at some stage during delivery, or after the user has received it (e.g.
11416 checking for tamper proof seals);
- 11417 c) observing that the process is applied in practise when the evaluator obtains the TOE
11418 through regular channels;
- 11419 d) questioning end users as to how the TOE was delivered.
- 11420 For guidance on site visits see A.4, Site Visits.

11421 It may be the case of a newly developed TOE that the delivery procedures have yet to be exercised.
 11422 In these cases, the evaluator has to be satisfied that appropriate procedures and facilities are in
 11423 place for future deliveries and that all personnel involved are aware of their responsibilities. The
 11424 evaluator may request a “dry run” of a delivery if this is practical. If the developer has produced
 11425 other similar products, then an examination of procedures in their use may be useful in providing
 11426 assurance.

11427 **13.5 Development security (ALC_DVS)**

11428 **13.5.1 Evaluation of sub-activity (ALC_DVS.1)**

11429 **13.5.1.1 Objectives**

11430 The objective of this sub-activity is to determine whether the developer's security controls on the
 11431 development environment are adequate to provide the confidentiality and integrity of the TOE
 11432 design and implementation that is necessary to ensure that secure operation of the TOE is not
 11433 compromised.

11434 **13.5.1.2 Input**

11435 The evaluation evidence for this sub-activity is:

- 11436 a) the ST;
- 11437 b) the development security documentation.

11438 In addition, the evaluator may need to examine other deliverables to determine that the security
 11439 controls are well-defined and followed. Specifically, the evaluator may need to examine the
 11440 developer's configuration management documentation (the input for the Evaluation of sub-activity
 11441 (ALC_CMC.4) “Production support and acceptance procedures” and the Evaluation of sub-activity
 11442 (ALC_CMS.4) “Problem tracking CM coverage”). Evidence that the procedures are being applied is
 11443 also required.

11444 **13.5.1.3 Action ALC_DVS.1.1E**

11445 ISO/IEC 15408-3 ALC_DVS.1.1C: *The development security documentation shall describe all the*
 11446 *physical, procedural, personnel, and other security measures that are necessary to protect the*
 11447 *confidentiality and integrity of the TOE design and implementation in its development environment.*

11448 **13.5.1.3.1 Work unit ALC_DVS.1-1**

11449 The evaluator **shall examine** the development security documentation to determine that it details
 11450 all security measures used in the development environment that are necessary to protect the
 11451 confidentiality and integrity of the TOE design and implementation.

11452 The evaluator determines what is necessary by first referring to the ST for any information that
 11453 may assist in the determination of necessary protection.

11454 If no explicit information is available from the ST the evaluator will need to make a determination
 11455 of the necessary measures. In cases where the developer's measures are considered less than what
 11456 is necessary, a clear justification should be provided for the assessment, based on a potential
 11457 exploitable vulnerability.

11458 The following types of security measures are considered by the evaluator when examining the
 11459 documentation:

- 11460 a) physical, for example physical access controls used to prevent unauthorised access to the
 11461 TOE development environment (during normal working hours and at other times);

- 11462 b) procedural, for example covering:
- 11463 • granting of access to the development environment or to specific parts of the environment
 - 11464 such as development machines
 - 11465 • revocation of access rights when a person leaves the development team
 - 11466 • transfer of protected material within and out of the development environment and between
 - 11467 different development sites in accordance with defined acceptance procedures
 - 11468 • admitting and escorting visitors to the development environment
 - 11469 • roles and responsibilities in ensuring the continued application of security measures, and
 - 11470 the detection of security breaches.
- 11471 c) personnel, for example any controls or checks made to establish the trustworthiness of
- 11472 new development staff;
- 11473 d) other security measures, for example the logical protections on any development
- 11474 machines.
- 11475 The development security documentation should identify the locations at which development
- 11476 occurs, and describe the aspects of development performed, along with the security measures
- 11477 applied at each location and for transports between different locations. For example, development
- 11478 could occur at multiple facilities within a single building, multiple buildings at the same site, or at
- 11479 multiple sites. Transports of parts of the TOE or the unfinished TOE between different
- 11480 development sites are to be covered by Development security (ALC_DVS), whereas the transport of
- 11481 the finished TOE to the consumer is dealt with in Delivery (ALC_DEL).
- 11482 Development includes the production of the TOE.
- 11483 **13.5.1.3.2 Work unit ALC_DVS.1-2**
- 11484 The evaluator ***shall examine*** the development confidentiality and integrity policies in order to
- 11485 determine the sufficiency of the security measures employed.
- 11486 The evaluator should examine whether the following is included in the policies:
- 11487 a) what information relating to the TOE development needs to be kept confidential, and
 - 11488 which members of the development staff are allowed to access such material;
 - 11489 b) what material must be protected from unauthorised modification in order to preserve the
 - 11490 integrity of the TOE, and which members of the development staff are allowed to modify
 - 11491 such material.
- 11492 The evaluator should determine that these policies are described in the development security
- 11493 documentation, that the security measures employed are consistent with the policies, and that they
- 11494 are complete.
- 11495 It should be noted that configuration management procedures will help protect the integrity of the
- 11496 TOE and the evaluator should avoid overlap with the work-units conducted for the CM capabilities
- 11497 (ALC_CMC). For example, the CM documentation may describe the security procedures necessary
- 11498 for controlling the roles or individuals who should have access to the development environment
- 11499 and who may modify the TOE.
- 11500 Whereas the CM capabilities (ALC_CMC) requirements are fixed, those for the Development
- 11501 security (ALC_DVS), mandating only necessary measures, are dependent on the nature of the TOE,

11502 and on information that may be provided in the ST. The evaluators would then determine that such
11503 a policy had been applied under this sub-activity.

11504 **13.5.1.4 Action ALC_DVS.1.2E**

11505 **13.5.1.4.1 Work unit ALC_DVS.1-3**

11506 The evaluator **shall examine** the development security documentation and associated evidence to
11507 determine that the security measures are being applied.

11508 This work unit requires the evaluator to determine that the security measures described in the
11509 development security documentation are being followed, such that the integrity of the TOE and the
11510 confidentiality of associated documentation is being adequately protected. For example, this could
11511 be determined by examination of the documentary evidence provided. Documentary evidence
11512 should be supplemented by visiting the development environment. A visit to the development
11513 environment will allow the evaluator to:

11514 a) observe the application of security measures (e.g. physical measures);

11515 b) examine documentary evidence of application of procedures;

11516 c) interview development staff to check awareness of the development security policies and
11517 procedures, and their responsibilities.

11518 A development site visit is a useful means of gaining confidence in the measures being used. Any
11519 decision not to make such a visit should be determined in consultation with the evaluation
11520 authority.

11521 For guidance on site visits see A.4, Site Visits.

11522 **13.5.2 Evaluation of sub-activity (ALC_DVS.2)**

11523 **13.5.2.1 Objectives**

11524 The objective of this sub-activity is to determine whether the developer's security controls on the
11525 development environment are adequate to provide the confidentiality and integrity of the TOE
11526 design and implementation that is necessary to ensure that secure operation of the TOE is not
11527 compromised. Additionally, sufficiency of the measures as applied is intended be justified.

11528 **13.5.2.2 Input**

11529 The evaluation evidence for this sub-activity is:

11530 a) the ST;

11531 b) the development security documentation.

11532 In addition, the evaluator may need to examine other deliverables to determine that the security
11533 controls are well-defined and followed. Specifically, the evaluator may need to examine the
11534 developer's configuration management documentation (the input for the Evaluation of sub-activity
11535 (ALC_CMC.4) "Production support and acceptance procedures" and the Evaluation of sub-activity
11536 (ALC_CMS.4) "Problem tracking CM coverage"). Evidence that the procedures are being applied is
11537 also required.

11538 **13.5.2.3 Action ALC_DVS.2.1E**

11539 ISO/IEC 15408-3 ALC_DVS.2.1C: *The development security documentation shall describe all the*
 11540 *physical, procedural, personnel, and other security measures that are necessary to protect the*
 11541 *confidentiality and integrity of the TOE design and implementation in its development environment.*

11542 **13.5.2.3.1 Work unit ALC_DVS.2-1**

11543 The evaluator ***shall examine*** the development security documentation to determine that it details
 11544 all security measures used in the development environment that are necessary to protect the
 11545 confidentiality and integrity of the TOE design and implementation.

11546 The evaluator determines what is necessary by first referring to the ST for any information that
 11547 may assist in the determination of necessary protection.

11548 If no explicit information is available from the ST the evaluator will need to make a determination
 11549 of the necessary measures. In cases where the developer's measures are considered less than what
 11550 is necessary, a clear justification should be provided for the assessment, based on a potential
 11551 exploitable vulnerability.

11552 The following types of security measures are considered by the evaluator when examining the
 11553 documentation:

- 11554 a) physical, for example physical access controls used to prevent unauthorised access to the
 11555 TOE development environment (during normal working hours and at other times);
- 11556 b) procedural, for example covering:
 - 11557 • granting of access to the development environment or to specific parts of the environment
 11558 such as development machines
 - 11559 • revocation of access rights when a person leaves the development team
 - 11560 • transfer of protected material out of the development environment and between different
 11561 development sites in accordance with defined acceptance procedures
 - 11562 • admitting and escorting visitors to the development environment
 - 11563 • roles and responsibilities in ensuring the continued application of security measures, and
 11564 the detection of security breaches.
- 11565 c) personnel, for example any controls or checks made to establish the trustworthiness of
 11566 new development staff;
- 11567 d) other security measures, for example the logical protections on any development
 11568 machines.

11569 The development security documentation should identify the locations at which development
 11570 occurs, and describe the aspects of development performed, along with the security measures
 11571 applied at each location and for transports between different locations. For example, development
 11572 could occur at multiple facilities within a single building, multiple buildings at the same site, or at
 11573 multiple sites. Transports of parts of the TOE or the unfinished TOE between different
 11574 development sites are to be covered by the Development security (ALC_DVS), whereas the
 11575 transport of the finished TOE to the consumer is dealt with in the Delivery (ALC_DEL).

11576 Development includes the production of the TOE.

11577 ISO/IEC 15408-3 ALC_DVS.2.2C: *The development security documentation shall justify that the*
11578 *security measures provide the necessary level of protection to maintain the confidentiality and*
11579 *integrity of the TOE.*

11580 **13.5.2.3.2 Work unit ALC_DVS.2-2**

11581 The evaluator ***shall examine*** the development security documentation to determine that an
11582 appropriate justification is given why the security measures provide the necessary level of
11583 protection to maintain the confidentiality and integrity of the TOE.

11584 Since attacks on the TOE or its related information are assumed in different design and production
11585 stages, measures and procedures need to have an appropriate level necessary to prevent those
11586 attacks or to make them more difficult.

11587 Since this level depends on the overall attack potential claimed for the TOE (cf. the Vulnerability
11588 analysis (AVA_VAN) component chosen), the development security documentation should justify
11589 the necessary level of protection to maintain the confidentiality and integrity of the TOE. This level
11590 has to be achieved by the security measures applied.

11591 The concept of protection measures should be consistent, and the justification should include an
11592 analysis of how the measures are mutually supportive. All aspects of development and production
11593 on all the different sites with all roles involved up to delivery of the TOE should be analysed.

11594 Justification may include an analysis of potential vulnerabilities taking the applied security
11595 measures into account.

11596 There may be a convincing argument showing that e.g.

11597 — The technical measures and mechanisms of the developer's infrastructure are sufficient for
11598 keeping the appropriate security level (e.g. cryptographic mechanisms as well as physical
11599 protection mechanisms, properties of the CM system (cf. ALC_CMC.4-5));

11600 — The system containing the implementation representation of the TOE (including concerning
11601 guidance documents) provides effective protection against logical attacks e.g. by "Trojan" code
11602 or viruses. It might be adequate, if the implementation representation is kept on an isolated
11603 system where only the software necessary to maintain it is installed and where no additional
11604 software is installed afterwards.

11605 — Data brought into this system need to be carefully considered to prevent the installation of
11606 hidden functionality onto the system. The effectiveness of these measures need to be tested, e.g.
11607 by independently trying to get access to the machine, install some additional executable
11608 (program, macro etc.) or get some information out of the machine using logical attacks.

11609 — The appropriate organisational (procedural and personal) measures are unconditionally
11610 enforced.

11611 **13.5.2.3.3 Work unit ALC_DVS.2-3**

11612 The evaluator ***shall examine*** the development confidentiality and integrity policies in order to
11613 determine the sufficiency of the security measures employed.

11614 The evaluator should examine whether the following is included in the policies:

11615 a) what information relating to the TOE development needs to be kept confidential, and
11616 which members of the development staff are allowed to access such material;

- 11617 b) what material must be protected from unauthorised modification in order to preserve the
 11618 integrity of the TOE, and which members of the development staff are allowed to modify
 11619 such material.
- 11620 The evaluator should determine that these policies are described in the development security
 11621 documentation, that the security measures employed are consistent with the policies, and that they
 11622 are complete.
- 11623 It should be noted that configuration management procedures will help protect the integrity of the
 11624 TOE and the evaluator should avoid overlap with the work-units conducted for the CM capabilities
 11625 (ALC_CMC). For example, the CM documentation may describe the security procedures necessary
 11626 for controlling the roles or individuals who should have access to the development environment
 11627 and who may modify the TOE.
- 11628 Whereas the CM capabilities (ALC_CMC) requirements are fixed, those for the Development
 11629 security (ALC_DVS), mandating only necessary measures, are dependent on the nature of the TOE,
 11630 and on information that may be provided in the ST. For example, the ST may identify a security
 11631 objective for the development environment that requires the TOE to be developed by staff that has
 11632 security clearance. The evaluators would then determine that such a policy had been applied under
 11633 this sub-activity.
- 11634 **13.5.2.4 Action ALC_DVS.2.2E**
- 11635 **13.5.2.4.1 Work unit ALC_DVS.2-4**
- 11636 The evaluator *shall examine* the development security documentation and associated evidence to
 11637 determine that the security measures are being applied.
- 11638 This work unit requires the evaluator to determine that the security measures described in the
 11639 development security documentation are being followed, such that the integrity of the TOE and the
 11640 confidentiality of associated documentation is being adequately protected. For example, this could
 11641 be determined by examination of the documentary evidence provided. Documentary evidence
 11642 should be supplemented by visiting the development environment. A visit to the development
 11643 environment will allow the evaluator to:
- 11644 a) observe the application of security measures (e.g. physical measures);
- 11645 b) examine documentary evidence of application of procedures;
- 11646 c) interview development staff to check awareness of the development security policies and
 11647 procedures, and their responsibilities.
- 11648 A development site visit is a useful means of gaining confidence in the measures being used. Any
 11649 decision not to make such a visit should be determined in consultation with the evaluation
 11650 authority.
- 11651 For guidance on site visits see A.4, Site Visits.
- 11652 **13.6 Flaw remediation (ALC_FLR)**
- 11653 **13.6.1 Evaluation of sub-activity (ALC_FLR.1)**
- 11654 **13.6.1.1 Objectives**
- 11655 The objective of this sub-activity is to determine whether the developer has established flaw
 11656 remediation procedures that describe the tracking of security flaws, the identification of corrective
 11657 actions, and the distribution of corrective action information to TOE users.

11658 **13.6.1.2 Input**

11659 The evaluation evidence for this sub-activity is:

11660 a) the flaw remediation procedures documentation.

11661 **13.6.1.3 Action ALC_FLR.1.1E**

11662 ISO/IEC 15408-3 ALC_FLR.1.1C: *The flaw remediation procedures documentation shall describe the*
11663 *procedures used to track all reported security flaws in each release of the TOE.*

11664 **13.6.1.3.1 Work unit ALC_FLR.1-1**

11665 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it
11666 describes the procedures used to track all reported security flaws in each release of the TOE.

11667 The procedures describe the actions that are taken by the developer from the time each suspected
11668 security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame,
11669 from initial detection through ascertaining that the flaw is a security flaw, to resolution of the
11670 security flaw.

11671 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw
11672 remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only
11673 that there be an explanation of why the flaw is not security-relevant.

11674 While these requirements do not mandate that there be a publicised means for TOE users to report
11675 security flaws, they do mandate that all security flaws that are reported be tracked. That is, a
11676 reported security flaw cannot be ignored simply because it comes from outside the developer's
11677 organisation.

11678 ISO/IEC 15408-3 ALC_FLR.1.2C: *The flaw remediation procedures shall require that a description of*
11679 *the nature and effect of each security flaw be provided, as well as the status of finding a correction to*
11680 *that flaw.*

11681 **13.6.1.3.2 Work unit ALC_FLR.1-2**

11682 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
11683 these procedures would produce a description of each security flaw in terms of its nature and
11684 effects.

11685 The procedures identify the actions that are taken by the developer to describe the nature and
11686 effects of each security flaw in sufficient detail to be able to reproduce it. The description of the
11687 nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design
11688 of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's
11689 effects identifies the portions of the TSF that are affected and how those portions are affected. For
11690 example, a security flaw in the implementation might be found that affects the identification and
11691 authentication enforced by the TSF by permitting authentication with the password "BACK DOOR".

11692 **13.6.1.3.3 Work unit ALC_FLR.1-3**

11693 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
11694 these procedures would identify the status of finding a correction to each security flaw.

11695 The flaw remediation procedures identify the different stages of security flaws. This differentiation
11696 includes at least: suspected security flaws that have been reported, suspected security flaws that
11697 have been confirmed to be security flaws, and security flaws whose solutions have been
11698 implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet

11699 investigated, flaws that are under investigation, security flaws for which a solution has been found
11700 but not yet implemented) be included.

11701 ISO/IEC 15408-3 ALC_FLR.1.3C: *The flaw remediation procedures shall require that corrective*
11702 *actions be identified for each of the security flaws.*

11703 **13.6.1.3.4 Work unit ALC_FLR.1-4**

11704 The evaluator **shall check** the flaw remediation procedures to determine that the application of
11705 these procedures would identify the corrective action for each security flaw.

11706 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the
11707 TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to
11708 TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes
11709 both those measures serving as only an interim solution (until the repair is issued) as well as those
11710 serving as a permanent solution (where it is determined that the procedural measure is the best
11711 solution).

11712 If the source of the security flaw is a documentation error, the corrective action consists of an
11713 update of the affected TOE guidance. If the corrective action is a procedural measure, this measure
11714 will include an update made to the affected TOE guidance to reflect these corrective procedures.

11715 ISO/IEC 15408-3 ALC_FLR.1.4C: *The flaw remediation procedures documentation shall describe the*
11716 *methods used to provide flaw information, corrections and guidance on corrective actions to TOE*
11717 *users.*

11718 **13.6.1.3.5 Work unit ALC_FLR.1-5**

11719 The evaluator **shall examine** the flaw remediation procedures documentation to determine that it
11720 describes a means of providing the TOE users with the necessary information on each security flaw.

11721 The *necessary information* about each security flaw consists of its description (not necessarily at
11722 the same level of detail as that provided as part of work unit ALC_FLR.1-2), the prescribed
11723 corrective action, and any associated guidance on implementing the correction.

11724 TOE users may be provided with such information, correction, and documentation updates in any
11725 of several ways, such as their posting to a website, their being sent to TOE users, or arrangements
11726 made for the developer to install the correction. In cases where the means of providing this
11727 information requires action to be initiated by the TOE user, the evaluator examines any TOE
11728 guidance to ensure that it contains instructions for retrieving the information.

11729 The only metric for assessing the adequacy of the method used for providing the information,
11730 corrections and guidance is that there be a reasonable expectation that TOE users can obtain or
11731 receive it. For example, consider the method of dissemination where the requisite data is posted to
11732 a website for one month, and the TOE users know that this will happen and when this will happen.
11733 This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet
11734 it is feasible that the TOE user could obtain the necessary information. On the other hand, if the
11735 information were posted to the website for only one hour, yet TOE users had no way of knowing
11736 this or when it would be posted, it is infeasible that they would ever get the necessary information.

11737 **13.6.2 Evaluation of sub-activity (ALC_FLR.2)**

11738 **13.6.2.1 Objectives**

11739 The objective of this sub-activity is to determine whether the developer has established flaw
11740 remediation procedures that describe the tracking of security flaws, the identification of corrective
11741 actions, and the distribution of corrective action information to TOE users. Additionally, this sub-
11742 activity determines whether the developer's procedures provide for the corrections of security

11743 flaws, for the receipt of flaw reports from TOE users, and for assurance that the corrections
11744 introduce no new security flaws.

11745 In order for the developer to be able to act appropriately upon security flaw reports from TOE
11746 users, TOE users need to understand how to submit security flaw reports to the developer, and
11747 developers need to know how to receive these reports. Flaw remediation guidance addressed to
11748 the TOE user ensures that TOE users are aware of how to communicate with the developer; flaw
11749 remediation procedures describe the developer's role in such communication

11750 **13.6.2.2 Input**

11751 The evaluation evidence for this sub-activity is:

11752 a) the flaw remediation procedures documentation;

11753 b) flaw remediation guidance documentation.

11754 **13.6.2.3 Action ALC_FLR.2.1E**

11755 ISO/IEC 15408-3 ALC_FLR.2.1C: *The flaw remediation procedures documentation shall describe the*
11756 *procedures used to track all reported security flaws in each release of the TOE.*

11757 **13.6.2.3.1 Work unit ALC_FLR.2-1**

11758 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it
11759 describes the procedures used to track all reported security flaws in each release of the TOE.

11760 The procedures describe the actions that are taken by the developer from the time each suspected
11761 security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame,
11762 from initial detection through ascertaining that the flaw is a security flaw, to resolution of the
11763 security flaw.

11764 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw
11765 remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only
11766 that there be an explanation of why the flaw is not security-relevant.

11767 ISO/IEC 15408-3 ALC_FLR.2.2C: *The flaw remediation procedures shall require that a description of*
11768 *the nature and effect of each security flaw be provided, as well as the status of finding a correction to*
11769 *that flaw.*

11770 **13.6.2.3.2 Work unit ALC_FLR.2-2**

11771 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
11772 these procedures would produce a description of each security flaw in terms of its nature and
11773 effects.

11774 The procedures identify the actions that are taken by the developer to describe the nature and
11775 effects of each security flaw in sufficient detail to be able to reproduce it. The description of the
11776 nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design
11777 of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's
11778 effects identifies the portions of the TSF that are affected and how those portions are affected. For
11779 example, a security flaw in the implementation might be found that affects the identification and
11780 authentication enforced by the TSF by permitting authentication with the password "BACKDOOR".

11781 **13.6.2.3.3 Work unit ALC_FLR.2-3**

11782 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
11783 these procedures would identify the status of finding a correction to each security flaw.

- 11784 The flaw remediation procedures identify the different stages of security flaws. This differentiation
 11785 includes at least: suspected security flaws that have been reported, suspected security flaws that
 11786 have been confirmed to be security flaws, and security flaws whose solutions have been
 11787 implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet
 11788 investigated, flaws that are under investigation, security flaws for which a solution has been found
 11789 but not yet implemented) be included.
- 11790 ISO/IEC 15408-3 ALC_FLR.2.3C: *The flaw remediation procedures shall require that corrective*
 11791 *actions be identified for each of the security flaws.*
- 11792 **13.6.2.3.4 Work unit ALC_FLR.2-4**
- 11793 The evaluator ***shall check*** the flaw remediation procedures to determine that the application of
 11794 these procedures would identify the corrective action for each security flaw.
- 11795 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the
 11796 TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to
 11797 TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes
 11798 both those measures serving as only an interim solution (until the repair is issued) as well as those
 11799 serving as a permanent solution (where it is determined that the procedural measure is the best
 11800 solution).
- 11801 If the source of the security flaw is a documentation error, the corrective action consists of an
 11802 update of the affected TOE guidance. If the corrective action is a procedural measure, this measure
 11803 will include an update made to the affected TOE guidance to reflect these corrective procedures.
- 11804 ISO/IEC 15408-3 ALC_FLR.2.4C: *The flaw remediation procedures documentation shall describe the*
 11805 *methods used to provide flaw information, corrections and guidance on corrective actions to TOE*
 11806 *users.*
- 11807 **13.6.2.3.5 Work unit ALC_FLR.2-5**
- 11808 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it
 11809 describes a means of providing the TOE users with the necessary information on each security flaw.
- 11810 *The necessary information* about each security flaw consists of its description (not necessarily at
 11811 the same level of detail as that provided as part of work unit ALC_FLR.2-2), the prescribed
 11812 corrective action, and any associated guidance on implementing the correction.
- 11813 TOE users may be provided with such information, correction, and documentation updates in any
 11814 of several ways, such as their posting to a website, their being sent to TOE users, or arrangements
 11815 made for the developer to install the correction. In cases where the means of providing this
 11816 information requires action to be initiated by the TOE user, the evaluator examines any TOE
 11817 guidance to ensure that it contains instructions for retrieving the information.
- 11818 The only metric for assessing the adequacy of the method used for providing the information,
 11819 corrections and guidance is that there be a reasonable expectation that TOE users can obtain or
 11820 receive it. For example, consider the method of dissemination where the requisite data is posted to
 11821 a website for one month, and the TOE users know that this will happen and when this will happen.
 11822 This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet
 11823 it is feasible that the TOE user could obtain the necessary information. On the other hand, if the
 11824 information were posted to the website for only one hour, yet TOE users had no way of knowing
 11825 this or when it would be posted, it is infeasible that they would ever get the necessary information.
- 11826 ISO/IEC 15408-3 ALC_FLR.2.5C: *The flaw remediation procedures shall describe a means by which*
 11827 *the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.*

11828 **13.6.2.3.6 Work unit ALC_FLR.2-6**

11829 The evaluator **shall examine** the flaw remediation procedures to determine that they describe
11830 procedures for the developer to accept reports of security flaws or requests for corrections to such
11831 flaws.

11832 The procedures ensure that TOE users have a means by which they can communicate with the TOE
11833 developer. By having a means of contact with the developer, the user can report security flaws,
11834 enquire about the status of security flaws, or request corrections to flaws. This means of contact
11835 may be part of a more general contact facility for reporting non-security related problems.

11836 The use of these procedures is not restricted to TOE users; however, only the TOE users are
11837 actively supplied with the details of these procedures. Others who might have access to or
11838 familiarity with the TOE can use the same procedures to submit reports to the developer, who is
11839 then expected to process them. Any means of submitting reports to the developer, other than those
11840 identified by the developer, are beyond the scope of this work unit; reports generated by other
11841 means need not be addressed.

11842 ISO/IEC 15408-3 ALC_FLR.2.6C: *The procedures for processing reported security flaws shall ensure*
11843 *that any reported flaws are remediated and the remediation procedures issued to TOE users.*

11844 **13.6.2.3.7 Work unit ALC_FLR.2-7**

11845 The evaluator **shall examine** the flaw remediation procedures to determine that the application of
11846 these procedures would help to ensure every reported flaw is corrected.

11847 The flaw remediation procedures cover not only those security flaws discovered and reported by
11848 developer personnel, but also those reported by TOE users. The procedures are sufficiently
11849 detailed so that they describe how it is ensured that each reported security flaw is corrected. The
11850 procedures contain reasonable steps that show progress leading to the eventual, inevitable
11851 resolution.

11852 The procedures describe the process that is taken from the point at which the suspected security
11853 flaw is determined to be a security flaw to the point at which it is resolved.

11854 **13.6.2.3.8 Work unit ALC_FLR.2-8**

11855 The evaluator **shall examine** the flaw remediation procedures to determine that the application of
11856 these procedures would help to ensure that the TOE users are issued remediation procedures for
11857 each security flaw.

11858 The procedures describe the process that is taken from the point at which a security flaw is
11859 resolved to the point at which the remediation procedures are provided. The procedures for
11860 delivering corrective actions should be consistent with the security objectives; they need not
11861 necessarily be identical to the procedures used for delivering the TOE, as documented to meet
11862 ALC_DEL, if included in the assurance requirements. For example, if the hardware portion of a TOE
11863 were originally delivered by bonded courier, updates to hardware resulting from flaw remediation
11864 would likewise be expected to be distributed by bonded courier. Updates unrelated to flaw
11865 remediation would follow the procedures set forth in the documentation meeting the Delivery
11866 (ALC_DEL) requirements.

11867 ISO/IEC 15408-3 ALC_FLR.2.7C: *The procedures for processing reported security flaws shall provide*
11868 *safeguards that any corrections to these security flaws do not introduce any new flaws.*

11869 **13.6.2.3.9 Work unit ALC_FLR.2-9**

11870 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 11871 these procedures would result in safeguards that the potential correction contains no adverse
 11872 effects.

11873 Through analysis, testing, or a combination of the two, the developer may reduce the likelihood
 11874 that adverse effects will be introduced when a security flaw is corrected. The evaluator assesses
 11875 whether the procedures provide detail in how the necessary mix of analysis and testing actions is
 11876 to be determined for a given correction.

11877 The evaluator also determines that, for instances where the source of the security flaw is a
 11878 documentation problem, the procedures include the means of safeguarding against the
 11879 introduction of contradictions with other documentation.

11880 ISO/IEC 15408-3 ALC_FLR.2.8C: *The flaw remediation guidance shall describe a means by which TOE*
 11881 *users report to the developer any suspected security flaws in the TOE.*

11882 **13.6.2.3.10 Work unit ALC_FLR.2-10**

11883 The evaluator ***shall examine*** the flaw remediation guidance to determine that the application of
 11884 these procedures would result in a means for the TOE user to provide reports of suspected security
 11885 flaws or requests for corrections to such flaws.

11886 The guidance ensures that TOE users have a means by which they can communicate with the TOE
 11887 developer. By having a means of contact with the developer, the user can report security flaws,
 11888 enquire about the status of security flaws, or request corrections to flaws.

11889 **13.6.3 Evaluation of sub-activity (ALC_FLR.3)**11890 **13.6.3.1 Objectives**

11891 The objective of this sub-activity is to determine whether the developer has established flaw
 11892 remediation procedures that describe the tracking of security flaws, the identification of corrective
 11893 actions, and the distribution of corrective action information to TOE users. Additionally, this sub-
 11894 activity determines whether the developer's procedures provide for the corrections of security
 11895 flaws, for the receipt of flaw reports from TOE users, for assurance that the corrections introduce
 11896 no new security flaws, for the establishment of a point of contact for each TOE user, and for the
 11897 timely issue of corrective actions to TOE users.

11898 In order for the developer to be able to act appropriately upon security flaw reports from TOE
 11899 users, TOE users need to understand how to submit security flaw reports to the developer, and
 11900 developers need to know how to receive these reports. Flaw remediation guidance addressed to
 11901 the TOE user ensures that TOE users are aware of how to communicate with the developer; flaw
 11902 remediation procedures describe the developer's role in such communication.

11903 **13.6.3.2 Input**

11904 The evaluation evidence for this sub-activity is:

11905 a) the flaw remediation procedures documentation;

11906 b) flaw remediation guidance documentation.

11907 **13.6.3.3 Action ALC_FLR.3.1E**

11908 ISO/IEC 15408-3 ALC_FLR.3.1C: *The flaw remediation procedures documentation shall describe the*
 11909 *procedures used to track all reported security flaws in each release of the TOE.*

11910 **13.6.3.3.1 Work unit ALC_FLR.3-1**

11911 The evaluator **shall examine** the flaw remediation procedures documentation to determine that it
11912 describes the procedures used to track all reported security flaws in each release of the TOE.

11913 The procedures describe the actions that are taken by the developer from the time each suspected
11914 security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame,
11915 from initial detection through ascertaining that the flaw is a security flaw, to resolution of the
11916 security flaw.

11917 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw
11918 remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only
11919 that there be an explanation of why the flaw is not security-relevant.

11920 ISO/IEC 15408-3 ALC_FLR.3.2C: *The flaw remediation procedures shall require that a description of*
11921 *the nature and effect of each security flaw be provided, as well as the status of finding a correction to*
11922 *that flaw.*

11923 **13.6.3.3.2 Work unit ALC_FLR.3-2**

11924 The evaluator **shall examine** the flaw remediation procedures to determine that the application of
11925 these procedures would produce a description of each security flaw in terms of its nature and
11926 effects.

11927 The procedures identify the actions that are taken by the developer to describe the nature and
11928 effects of each security flaw in sufficient detail to be able to reproduce it. The description of the
11929 nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design
11930 of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's
11931 effects identifies the portions of the TSF that are affected and how those portions are affected. For
11932 example, a security flaw in the implementation might be found that affects the identification and
11933 authentication enforced by the TSF by permitting authentication with the password "BACKDOOR".

11934 **13.6.3.3.3 Work unit ALC_FLR.3-3**

11935 The evaluator **shall examine** the flaw remediation procedures to determine that the application of
11936 these procedures would identify the status of finding a correction to each security flaw.

11937 The flaw remediation procedures identify the different stages of security flaws. This differentiation
11938 includes at least: suspected security flaws that have been reported, suspected security flaws that
11939 have been confirmed to be security flaws, and security flaws whose solutions have been
11940 implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet
11941 investigated, flaws that are under investigation, security flaws for which a solution has been found
11942 but not yet implemented) be included.

11943 ISO/IEC 15408-3 ALC_FLR.3.3C: *The flaw remediation procedures shall require that corrective*
11944 *actions be identified for each of the security flaws.*

11945 **13.6.3.3.4 Work unit ALC_FLR.3-4**

11946 The evaluator **shall check** the flaw remediation procedures to determine that the application of
11947 these procedures would identify the corrective action for each security flaw.

11948 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the
11949 TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to
11950 TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes
11951 both those measures serving as only an interim solution (until the repair is issued) as well as those
11952 serving as a permanent solution (where it is determined that the procedural measure is the best
11953 solution).

- 11954 If the source of the security flaw is a documentation error, the corrective action consists of an
 11955 update of the affected TOE guidance. If the corrective action is a procedural measure, this measure
 11956 will include an update made to the affected TOE guidance to reflect these corrective procedures.
- 11957 ISO/IEC 15408-3 ALC_FLR.3.4C: *The flaw remediation procedures documentation shall describe the*
 11958 *methods used to provide flaw information, corrections and guidance on corrective actions to TOE*
 11959 *users.*
- 11960 **13.6.3.3.5 Work unit ALC_FLR.3-5**
- 11961 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it
 11962 describes a means of providing the TOE users with the necessary information on each security flaw.
- 11963 *The necessary information* about each security flaw consists of its description (not necessarily at
 11964 the same level of detail as that provided as part of work unit ALC_FLR.3-2), the prescribed
 11965 corrective action, and any associated guidance on implementing the correction.
- 11966 TOE users may be provided with such information, correction, and documentation updates in any
 11967 of several ways, such as their posting to a website, their being sent to TOE users, or arrangements
 11968 made for the developer to install the correction. In cases where the means of providing this
 11969 information requires action to be initiated by the TOE user, the evaluator examines any TOE
 11970 guidance to ensure that it contains instructions for retrieving the information.
- 11971 The only metric for assessing the adequacy of the method used for providing the information,
 11972 corrections and guidance is that there be a reasonable expectation that TOE users can obtain or
 11973 receive it. For example, consider the method of dissemination where the requisite data is posted to
 11974 a website for one month, and the TOE users know that this will happen and when this will happen.
 11975 This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet
 11976 it is feasible that the TOE user could obtain the necessary information. On the other hand, if the
 11977 information were posted to the website for only one hour, yet TOE users had no way of knowing
 11978 this or when it would be posted, it is infeasible that they would ever get the necessary information.
- 11979 For TOE users who register with the developer (see work unit ALC_FLR.3-12), the passive
 11980 availability of this information is not sufficient. Developers must actively send the information (or a
 11981 notification of its availability) to registered TOE users.
- 11982 ISO/IEC 15408-3 ALC_FLR.3.5C: *The flaw remediation procedures shall describe a means by which*
 11983 *the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.*
- 11984 **13.6.3.3.6 Work unit ALC_FLR.3-6**
- 11985 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 11986 these procedures would result in a means for the developer to receive from TOE user reports of
 11987 suspected security flaws or requests for corrections to such flaws.
- 11988 The procedures ensure that TOE users have a means by which they can communicate with the TOE
 11989 developer. By having a means of contact with the developer, the user can report security flaws,
 11990 enquire about the status of security flaws, or request corrections to flaws. This means of contact
 11991 may be part of a more general contact facility for reporting non-security related problems.
- 11992 The use of these procedures is not restricted to TOE users; however, only the TOE users are
 11993 actively supplied with the details of these procedures. Others who might have access to or
 11994 familiarity with the TOE can use the same procedures to submit reports to the developer, who is
 11995 then expected to process them. Any means of submitting reports to the developer, other than those
 11996 identified by the developer, are beyond the scope of this work unit; reports generated by other
 11997 means need not be addressed.

11998 ISO/IEC 15408-3 ALC_FLR.3.6C: *The flaw remediation procedures shall include a procedure*
 11999 *requiring timely response and the automatic distribution of security flaw reports and the associated*
 12000 *corrections to registered users who might be affected by the security flaw.*

12001 **13.6.3.3.7 Work unit ALC_FLR.3-7**

12002 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 12003 these procedures would result in a timely means of providing the registered TOE users who might
 12004 be affected with reports about, and associated corrections to, each security flaw.

12005 The issue of timeliness applies to the issuance of both security flaw reports and the associated
 12006 corrections. However, these need not be issued at the same time. It is recognised that flaw reports
 12007 should be generated and issued as soon as an interim solution is found, even if that solution is as
 12008 drastic as turn off the TOE. Likewise, when a more permanent (and less drastic) solution is found, it
 12009 should be issued without undue delay.

12010 It is unnecessary to restrict the recipients of the reports and associated corrections to only those
 12011 TOE users who might be affected by the security flaw; it is permissible that all TOE users be given
 12012 such reports and corrections for all security flaws, provided such is done in a timely manner.

12013 **13.6.3.3.8 Work unit ALC_FLR.3-8**

12014 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 12015 these procedures would result in automatic distribution of the reports and associated corrections
 12016 to the registered TOE users who might be affected.

12017 *Automatic distribution* does not mean that human interaction with the distribution method is not
 12018 permitted. In fact, the distribution method could consist entirely of manual procedures, perhaps
 12019 through a closely monitored procedure with prescribed escalation upon the lack of issue of reports
 12020 or corrections.

12021 It is unnecessary to restrict the recipients of the reports and associated corrections to only those
 12022 TOE users who might be affected by the security flaw; it is permissible that all TOE users be given
 12023 such reports and corrections for all security flaws, provided such is done automatically.

12024 ISO/IEC 15408-3 ALC_FLR.3.7C: *The procedures for processing reported security flaws shall ensure*
 12025 *that any reported flaws are remediated and the remediation procedures issued to TOE users.*

12026 **13.6.3.3.9 Work unit ALC_FLR.3-9**

12027 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 12028 these procedures would help to ensure that every reported flaw is corrected.

12029 The flaw remediation procedures cover not only those security flaws discovered and reported by
 12030 developer personnel, but also those reported by TOE users. The procedures are sufficiently
 12031 detailed so that they describe how it is ensured that each reported security flaw is remediated. The
 12032 procedures contain reasonable steps that show progress leading to the eventual, inevitable
 12033 resolution.

12034 The procedures describe the process that is taken from the point at which the suspected security
 12035 flaw is determined to be a security flaw to the point at which it is resolved.

12036 **13.6.3.3.10 Work unit ALC_FLR.3-10**

12037 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 12038 these procedures would help to ensure that the TOE users are issued remediation procedures for
 12039 each security flaw.

- 12040 The procedures describe the process that is taken from the point at which a security flaw is
 12041 resolved to the point at which the remediation procedures are provided. The procedures for
 12042 delivering remediation procedures should be consistent with the security objectives; they need not
 12043 necessarily be identical to the procedures used for delivering the TOE, as documented to meet
 12044 Delivery (ALC_DEL), if included in the assurance requirements. For example, if the hardware
 12045 portion of a TOE were originally delivered by bonded courier, updates to hardware resulting from
 12046 flaw remediation would likewise be expected to be distributed by bonded courier. Updates
 12047 unrelated to flaw remediation would follow the procedures set forth in the documentation meeting
 12048 the Delivery (ALC_DEL) requirements.
- 12049 ISO/IEC 15408-3 ALC_FLR.3.8C: *The procedures for processing reported security flaws shall provide*
 12050 *safeguards that any corrections to these security flaws do not introduce any new flaws.*
- 12051 **13.6.3.3.11 Work unit ALC_FLR.3-11**
- 12052 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of
 12053 these procedures would result in safeguards that the potential correction contains no adverse
 12054 effects.
- 12055 Through analysis, testing, or a combination of the two, the developer may reduce the likelihood
 12056 that adverse effects will be introduced when a security flaw is corrected. The evaluator assesses
 12057 whether the procedures provide detail in how the necessary mix of analysis and testing actions is
 12058 to be determined for a given correction.
- 12059 The evaluator also determines that, for instances where the source of the security flaw is a
 12060 documentation problem, the procedures include the means of safeguarding against the
 12061 introduction of contradictions with other documentation.
- 12062 ISO/IEC 15408-3 ALC_FLR.3.9C: *The flaw remediation guidance shall describe a means by which TOE*
 12063 *users report to the developer any suspected security flaws in the TOE.*
- 12064 **13.6.3.3.12 Work unit ALC_FLR.3-12**
- 12065 The evaluator ***shall examine*** the flaw remediation guidance to determine that the application of
 12066 these procedures would result in a means for the TOE user to provide reports of suspected security
 12067 flaws or requests for corrections to such flaws.
- 12068 The guidance ensures that TOE users have a means by which they can communicate with the TOE
 12069 developer. By having a means of contact with the developer, the user can report security flaws,
 12070 enquire about the status of security flaws, or request corrections to flaws.
- 12071 ISO/IEC 15408-3 ALC_FLR.3.10C: *The flaw remediation guidance shall describe a means by which*
 12072 *TOE users may register with the developer, to be eligible to receive security flaw reports and*
 12073 *corrections.*
- 12074 **13.6.3.3.13 Work unit ALC_FLR.3-13**
- 12075 The evaluator ***shall examine*** the flaw remediation guidance to determine that it describes a means
 12076 of enabling the TOE users to register with the developer.
- 12077 *Enabling the TOE users to register with the developer* simply means having a way for each TOE user
 12078 to provide the developer with a point of contact; this point of contact is to be used to provide the
 12079 TOE user with information related to security flaws that might affect that TOE user, along with any
 12080 corrections to the security flaw. Registering the TOE user may be accomplished as part of the
 12081 standard procedures that TOE users undergo to identify themselves to the developer, for the
 12082 purposes of registering a software licence, or for obtaining update and other useful information.

12083 There need not be one registered TOE user per installation of the TOE; it would be sufficient if
 12084 there were one registered TOE user for an organisation. For example, a corporate TOE user might
 12085 have a centralised acquisition office for all of its sites. In this case, the acquisition office would be a
 12086 sufficient point of contact for all of that TOE user's sites, so that all of the TOE user's installations of
 12087 the TOE have a registered point of contact.

12088 In either case, it must be possible to associate each TOE that is delivered with an organisation in
 12089 order to ensure that there is a registered user for each TOE. For organisations that have many
 12090 different addresses, this assures that there will be no user who is erroneously presumed to be
 12091 covered by a registered TOE user.

12092 It should be noted that TOE users need not register; they must only be provided with a means of
 12093 doing so. However, users who choose to register must be directly sent the information (or a
 12094 notification of its availability).

12095 ISO/IEC 15408-3 ALC_FLR.3.11C: *The flaw remediation guidance shall identify the specific points of*
 12096 *contact for all reports and enquiries about security issues involving the TOE.*

12097 **13.6.3.3.14 Work unit ALC_FLR.3-14**

12098 The evaluator ***shall examine*** the flaw remediation guidance to determine that it identifies specific
 12099 points of contact for user reports and enquiries about security issues involving the TOE.

12100 The guidance includes a means whereby registered TOE users can interact with the developer to
 12101 report discovered security flaws in the TOE or to make enquiries regarding discovered security
 12102 flaws in the TOE.

12103 **13.7 Life-cycle definition (ALC_LCD)**

12104 **13.7.1 Evaluation of sub-activity (ALC_LCD.1)**

12105 **13.7.1.1 Objectives**

12106 The objective of this sub-activity is to determine whether the developer has used a documented
 12107 model of the TOE life-cycle.

12108 **13.7.1.2 Input**

12109 The evaluation evidence for this sub-activity is:

- 12110 a) the ST;
- 12111 b) the life-cycle definition documentation.

12112 **13.7.1.3 Action ALC_LCD.1.1E**

12113 ISO/IEC 15408-3 ALC_LCD.1.1C: *The life-cycle definition documentation shall describe the model*
 12114 *used to develop and maintain the TOE.*

12115 **13.7.1.3.1 Work unit ALC_LCD.1-1**

12116 The evaluator ***shall examine*** the documented description of the life-cycle model used to determine
 12117 that it covers the development and maintenance process.

12118 The description of the life-cycle model should include:

- 12119 a) information on the life-cycle phases of the TOE and the boundaries between the
 12120 subsequent phases;

- 12121 b) information on the procedures, tools and techniques used by the developer (e.g. for
12122 design, coding, testing, bug-fixing);
- 12123 c) overall management structure governing the application of the procedures (e.g. an
12124 identification and description of the individual responsibilities for each of the procedures
12125 required by the development and maintenance process covered by the life-cycle model);
- 12126 d) information on which parts of the TOE are delivered by subcontractors, if subcontractors
12127 are involved.
- 12128 Evaluation of sub-activity (ALC_LCD.1) does not require the model used to conform to any standard
12129 life-cycle model.
- 12130 ISO/IEC 15408-3 ALC_LCD.1.2C: *The life-cycle model shall provide for the necessary control over the*
12131 *development and maintenance of the TOE.*
- 12132 **13.7.1.3.2 Work unit ALC_LCD.1-2**
- 12133 The evaluator ***shall examine*** the life-cycle model to determine that use of the procedures, tools
12134 and techniques described by the life-cycle model will make the necessary positive contribution to
12135 the development and maintenance of the TOE.
- 12136 The information provided in the life-cycle model gives the evaluator assurance that the
12137 development and maintenance procedures adopted would minimise the likelihood of security
12138 flaws. For example, if the life-cycle model described the review process, but did not make provision
12139 for recording changes to components, then the evaluator may be less confident that errors will not
12140 be introduced into the TOE. The evaluator may gain further assurance by comparing the
12141 description of the model against an understanding of the development process gleaned from
12142 performing other evaluator actions relating to the TOE development (e.g. those covered under the
12143 CM capabilities (ALC_CMC)). Identified deficiencies in the life-cycle model will be of concern if they
12144 might reasonably be expected to give rise to the introduction of flaws into the TOE, either
12145 accidentally or deliberately.
- 12146 ISO/IEC 15408 does not mandate any particular development approach, and each should be judged
12147 on merit. For example, spiral, rapid-prototyping and waterfall approaches to design can all be used
12148 to produce a quality TOE if applied in a controlled environment.
- 12149 **13.7.2 Evaluation of sub-activity (ALC_LCD.2)**
- 12150 **13.7.2.1 Objectives**
- 12151 The objective of this sub-activity is to determine whether the developer has used a documented
12152 and measurable model of the TOE life-cycle.
- 12153 **13.7.2.2 Input**
- 12154 The evaluation evidence for this sub-activity is:
- 12155 a) the ST;
- 12156 b) the life-cycle definition documentation;
- 12157 c) information about the standard used;
- 12158 d) the life-cycle output documentation.

12159 **13.7.2.3 Action ALC_LCD.2.1E**

12160 ISO/IEC 15408-3 ALC_LCD.2.1C: *The life-cycle definition documentation shall describe the model*
 12161 *used to develop and maintain the TOE, including the details of its arithmetic parameters and/or*
 12162 *metrics used to measure the quality of the TOE and/or its development.*

12163 **13.7.2.3.1 Work unit ALC_LCD.2-1**

12164 The evaluator **shall examine** the documented description of the life-cycle model used to determine
 12165 that it covers the development and maintenance process, including the details of its arithmetic
 12166 parameters and/or metrics used to measure the TOE development.

12167 The description of the life-cycle model includes:

12168 a) information on the life-cycle phases of the TOE and the boundaries between the
 12169 subsequent phases;

12170 b) information on the procedures, tools and techniques used by the developer (e.g. for
 12171 design, coding, testing, bug-fixing);

12172 c) overall management structure governing the application of the procedures (e.g. an
 12173 identification and description of the individual responsibilities for each of the procedures
 12174 required by the development and maintenance process covered by the life-cycle model);

12175 d) information on which parts of the TOE are delivered by subcontractors, if subcontractors
 12176 are involved;

12177 e) information on the parameters/metrics that are used to measure the TOE development.
 12178 Metrics standards typically include guides for measuring and producing reliable products
 12179 and cover the aspects reliability, quality, performance, complexity and cost. For the
 12180 evaluation all those metrics are of relevance, which are used to increase quality by
 12181 decreasing the probability of faults and thereby in turn increase assurance in the security
 12182 of the TOE.

12183 ISO/IEC 15408-3 ALC_LCD.2.2C: *The life-cycle model shall provide for the necessary control over the*
 12184 *development and maintenance of the TOE.*

12185 **13.7.2.3.2 Work unit ALC_LCD.2-2**

12186 The evaluator **shall examine** the life-cycle model to determine that use of the procedures, tools
 12187 and techniques described by the life-cycle model will make the necessary positive contribution to
 12188 the development and maintenance of the TOE.

12189 The information provided in the life-cycle model gives the evaluator assurance that the
 12190 development and maintenance procedures adopted would minimise the likelihood of security
 12191 flaws. For example, if the life-cycle model described the review process, but did not make provision
 12192 for recording changes to components, then the evaluator may be less confident that errors will not
 12193 be introduced into the TOE. The evaluator may gain further assurance by comparing the
 12194 description of the model against an understanding of the development process gleaned from
 12195 performing other evaluator actions relating to the TOE development (e.g. those covered under the
 12196 CM capabilities (ALC_CMC)). Identified deficiencies in the life-cycle model will be of concern if they
 12197 might reasonably be expected to give rise to the introduction of flaws into the TOE, either
 12198 accidentally or deliberately.

12199 ISO/IEC 15408 does not mandate any particular development approach, and each should be judged
 12200 on merit. For example, spiral, rapid-prototyping and waterfall approaches to design can all be used
 12201 to produce a quality TOE if applied in a controlled environment.

12202 For the metrics/measurements used in the life-cycle model, evidence has to be provided that
 12203 shows how those metrics/measurements usefully contribute to the minimisation of the likelihood
 12204 of flaws. This can be viewed as the overall goal for measurement in an ALC context. As a
 12205 consequence the metrics/measurements have to be selected based on their capability to achieve
 12206 that overall goal or contribute to that. In the first place a metric/measure is suitable with respect to
 12207 ALC if a correlation between the metric/measure and the number of flaws can be stated with a
 12208 certain degree of reliability. But also a metric/measure useful for management purposes as for
 12209 planning and monitoring the TOE development are helpful since badly managed projects are
 12210 endangered to produce bad quality and to introduce flaws.

12211 It may be possible to use metrics for quality improvement, for which this use is not obvious. For
 12212 example a metric to estimate the expected cost of a product development may help quality, if the
 12213 developer can show that this is used to provide an adequate budget for development projects and
 12214 that this helps to avoid quality problems arising from resource shortages.

12215 It is not required that every single step in the life cycle of the TOE is measurable. However the
 12216 evaluator should see from the description of the measures and procedures that the metrics are
 12217 appropriate to control the overall quality of the TOE and to minimise possible security flaws by this.

12218 ISO/IEC 15408-3 ALC_LCD.2.3C: *The life-cycle output documentation shall provide the results of the*
 12219 *measurements of the TOE development using the measurable life-cycle model.*

12220 **13.7.2.3.3 Work unit ALC_LCD.2-3**

12221 The evaluator ***shall examine*** the life-cycle output documentation to determine that it provides the
 12222 results of the measurements of the TOE development using the measurable life-cycle model.

12223 The results of the measurements and the life-cycle progress of the TOE should be in accordance
 12224 with the life-cycle model.

12225 The output documentation not only includes numeric values of the metrics but also documents
 12226 actions taken as a result of the measurements and in accordance with the model. For example there
 12227 may be a requirement that a certain design phase needs to be repeated, if some error rates
 12228 measured during testing are outside of a defined threshold. In this case the documentation should
 12229 show that such action was taken, if indeed the thresholds were not met.

12230 If the evaluation is conducted in parallel with the development of the TOE it may be possible that
 12231 quality measurements have not been used in the past. In this case the evaluator should use the
 12232 documentation of the planned procedures in order to gain confidence that corrective actions are
 12233 defined if results of quality measurements deviate from some threshold.

12234 **13.8 Tools and techniques (ALC_TAT)**

12235 **13.8.1 Evaluation of sub-activity (ALC_TAT.1)**

12236 **13.8.1.1 Objectives**

12237 The objective of this sub-activity is to determine whether the developer has used well-defined
 12238 development tools (e.g. programming languages or computer-aided design (CAD) systems) that
 12239 yield consistent and predictable results.

12240 **13.8.1.2 Input**

12241 The evaluation evidence for this sub-activity is:

- 12242 a) the development tool documentation;
- 12243 b) the subset of the implementation representation.

12244 **13.8.1.3 Application notes**

12245 This work may be performed in parallel with the evaluation activities under Implementation
12246 representation (ADV_IMP), specifically with regard to determining the use of features in the tools
12247 that will affect the object code (e.g. compilation options).

12248 **13.8.1.4 Action ALC_TAT.1.1E**

12249 ISO/IEC 15408-3 ALC_TAT.1.1C: *Each development tool used for implementation shall be well-*
12250 *defined.*

12251 **13.8.1.4.1 Work unit ALC_TAT.1-1**

12252 The evaluator ***shall examine*** the development tool documentation provided to determine that
12253 each development tools is well-defined.

12254 For example, a well-defined language, compiler or CAD system may be considered to be one that
12255 conforms to a recognised standard, such as the ISO standards. A well-defined language is one that
12256 has a clear and complete description of its syntax, and a detailed description of the semantics of
12257 each construct.

12258 ISO/IEC 15408-3 ALC_TAT.1.2C: *The documentation of each development tool shall unambiguously*
12259 *define the meaning of all statements as well as all conventions and directives used in the*
12260 *implementation.*

12261 **13.8.1.4.2 Work unit ALC_TAT.1-2**

12262 The evaluator ***shall examine*** the documentation of each development tool to determine that it
12263 unambiguously defines the meaning of all statements as well as all conventions and directives used
12264 in the implementation.

12265 The development tool documentation (e.g. programming language specifications and user
12266 manuals) should cover all statements used in the implementation representation of the TOE, and
12267 for each such statement should provide a clear and unambiguous definition of the purpose and
12268 effect of that statement. This work may be performed in parallel with the evaluator's examination
12269 of the implementation representation performed during the ADV_IMP sub-activity. The key test the
12270 evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to
12271 be able to understand the implementation representation. The documentation should not assume
12272 (for example) that the reader is an expert in the programming language used.

12273 Reference to the use of a documented standard is an acceptable approach to meet this requirement,
12274 provided that the standard is available to the evaluator. Any differences from the standard should
12275 be documented.

12276 The critical test is whether the evaluator can understand the TOE source code when performing
12277 source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can
12278 additionally be used in searching for problem areas:

- 12279 a) In the language definition, phrases such as “the effect of this construct is undefined” and
12280 terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.
- 12281 b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a
12282 common source of ambiguity problems.
- 12283 c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is
12284 often poorly defined.

- 12285 Most languages in common use, however well designed, will have some problematic constructs. If
 12286 the implementation language is mostly well defined, but some problematic constructs exist, then
 12287 an inconclusive verdict should be assigned, pending examination of the source code.
- 12288 The evaluator should verify, during the examination of source code, that any use of the problematic
 12289 constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs
 12290 precluded by the documented standard are not used.
- 12291 The development tool documentation should define all conventions and directives used in the
 12292 implementation.
- 12293 ISO/IEC 15408-3 ALC_TAT.1.3C: *The documentation of each development tool shall unambiguously*
 12294 *define the meaning of all implementation-dependent options.*
- 12295 **13.8.1.4.3 Work unit ALC_TAT.1-3**
- 12296 The evaluator ***shall examine*** the development tool documentation to determine that it
 12297 unambiguously defines the meaning of all implementation-dependent options.
- 12298 The documentation of software development tools should include definitions of implementation-
 12299 dependent options that may affect the meaning of the executable code, and those that are different
 12300 from the standard language as documented. Where source code is provided to the evaluator,
 12301 information should also be provided on compilation and linking options used.
- 12302 The documentation for hardware design and development tools should describe the use of all
 12303 options that affect the output from the tools (e.g. detailed hardware specifications, or actual
 12304 hardware).
- 12305 **13.8.2 Evaluation of sub-activity (ALC_TAT.2)**
- 12306 **13.8.2.1 Objectives**
- 12307 The objective of this sub-activity is to determine whether the developer has used well-defined
 12308 development tools (e.g. programming languages or computer-aided design (CAD) systems) that
 12309 yield consistent and predictable results, and whether implementation standards have been applied.
- 12310 **13.8.2.2 Input**
- 12311 The evaluation evidence for this sub-activity is:
- 12312 a) the development tool documentation;
- 12313 b) the implementation standards description;
- 12314 c) the provided implementation representation of the TSF.
- 12315 **13.8.2.3 Application notes**
- 12316 This work may be performed in parallel with the evaluation activities under ADV_IMP, specifically
 12317 with regard to determining the use of features in the tools that will affect the object code (e.g.
 12318 compilation options).
- 12319 **13.8.2.4 Action ALC_TAT.2.1E**
- 12320 ISO/IEC 15408-3 ALC_TAT.2.1C: *Each development tool used for implementation shall be well-*
 12321 *defined.*

12322 **13.8.2.4.1 Work unit ALC_TAT.2-1**

12323 The evaluator **shall examine** the development tool documentation provided to determine that
12324 each development tool is well-defined.

12325 For example, a well-defined language, compiler or CAD system may be considered to be one that
12326 conforms to a recognised standard, such as the ISO standards. A well-defined language is one that
12327 has a clear and complete description of its syntax, and a detailed description of the semantics of
12328 each construct.

12329 ISO/IEC 15408-3 ALC_TAT.2.2C: *The documentation of each development tool shall unambiguously*
12330 *define the meaning of all statements as well as all conventions and directives used in the*
12331 *implementation.*

12332 **13.8.2.4.2 Work unit ALC_TAT.2-2**

12333 The evaluator **shall examine** the documentation of each development tool to determine that it
12334 unambiguously defines the meaning of all statements as well as all conventions and directives used
12335 in the implementation.

12336 The development tool documentation (e.g. programming language specifications and user
12337 manuals) should cover all statements used in the implementation representation of the TOE, and
12338 for each such statement should provide a clear and unambiguous definition of the purpose and
12339 effect of that statement. This work may be performed in parallel with the evaluator's examination
12340 of the implementation representation performed during the ADV_IMP sub-activity. The key test the
12341 evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to
12342 be able to understand the implementation representation. The documentation should not assume
12343 (for example) that the reader is an expert in the programming language used.

12344 Reference to the use of a documented standard is an acceptable approach to meet this requirement,
12345 provided that the standard is available to the evaluator. Any differences from the standard should
12346 be documented.

12347 The critical test is whether the evaluator can understand the TOE source code when performing
12348 source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can
12349 additionally be used in searching for problem areas:

12350 a) In the language definition, phrases such as “the effect of this construct is undefined” and
12351 terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.

12352 b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a
12353 common source of ambiguity problems.

12354 c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is
12355 often poorly defined.

12356 Most languages in common use, however well designed, will have some problematic constructs. If
12357 the implementation language is mostly well defined, but some problematic constructs exist, then
12358 an inconclusive verdict should be assigned, pending examination of the source code.

12359 The evaluator should verify, during the examination of source code, that any use of the problematic
12360 constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs
12361 precluded by the documented standard are not used.

12362 The development tool documentation should define all conventions and directives used in the
12363 implementation.

12364 ISO/IEC 15408-3 ALC_TAT.2.3C: *The documentation of each development tool shall unambiguously*
 12365 *define the meaning of all implementation-dependent options.*

12366 **13.8.2.4.3 Work unit ALC_TAT.2-3**

12367 The evaluator ***shall examine*** the development tool documentation to determine that it
 12368 unambiguously defines the meaning of all implementation-dependent options.

12369 The documentation of software development tools should include definitions of implementation-
 12370 dependent options that may affect the meaning of the executable code, and those that are different
 12371 from the standard language as documented. Where source code is provided to the evaluator,
 12372 information should also be provided on compilation and linking options used.

12373 The documentation for hardware design and development tools should describe the use of all
 12374 options that affect the output from the tools (e.g. detailed hardware specifications, or actual
 12375 hardware).

12376 **13.8.2.5 Action ALC_TAT.2.2E**

12377 **13.8.2.5.1 Work unit ALC_TAT.2-4**

12378 The evaluator ***shall examine*** aspects of the implementation process to determine that documented
 12379 implementation standards have been applied.

12380 This work unit requires the evaluator to analyse the provided implementation representation of
 12381 the TOE to determine whether the documented implementation standards have been applied.

12382 The evaluator should verify that constructs excluded by the documented standard are not used.

12383 Additionally, the evaluator should verify the developer's procedures which ensure the application
 12384 of the defined standards within the design and implementation process of the TOE. Therefore,
 12385 documentary evidence should be supplemented by visiting the development environment. A visit
 12386 to the development environment will allow the evaluator to:

12387 a) observe the application of defined standards;

12388 b) examine documentary evidence of application of procedures describing the use of defined
 12389 standards;

12390 c) interview development staff to check awareness of the application of defined standards
 12391 and procedures.

12392 A development site visit is a useful means of gaining confidence in the procedures being used. Any
 12393 decision not to make such a visit should be determined in consultation with the evaluation
 12394 authority.

12395 The evaluator compares the provided implementation representation with the description of the
 12396 applied implementation standards and verifies their use.

12397 At this level it is not required that the complete provided implementation representation of the
 12398 TSF is based on implementation standards, but only those parts that are developed by the TOE
 12399 developer himself. The evaluator may consult the configuration list required by the CM scope
 12400 (ALC_CMS) to get the information which parts are developed by the TOE developer, and which by
 12401 third party developers.

12402 If the referenced implementation standards are not applied for at least parts of the provided
 12403 implementation representation, the evaluator action related to this work unit is assigned a fail
 12404 verdict.

- 12405 Note that parts of the TOE which are not TSF relevant do not need to be examined.
- 12406 This work unit may be performed in conjunction with the evaluation activities under ADV_IMP.
- 12407 **13.8.3 Evaluation of sub-activity (ALC_TAT.3)**
- 12408 **13.8.3.1 Objectives**
- 12409 The objective of this sub-activity is to determine whether the developer and their subcontractors
12410 have used well-defined development tools (e.g. programming languages or computer-aided design
12411 (CAD) systems) that yield consistent and predictable results, and whether implementation
12412 standards have been applied.
- 12413 **13.8.3.2 Input**
- 12414 The evaluation evidence for this sub-activity is:
- 12415 a) the development tool documentation;
- 12416 b) the implementation standards description;
- 12417 c) the provided implementation representation of the TSF.
- 12418 **13.8.3.3 Application notes**
- 12419 This work may be performed in parallel with the evaluation activities under ADV_IMP, specifically
12420 with regard to determining the use of features in the tools that will affect the object code (e.g.
12421 compilation options).
- 12422 **13.8.3.4 Action ALC_TAT.3.1E**
- 12423 ISO/IEC 15408-3 ALC_TAT.3.1C: *Each development tool used for implementation shall be well-*
12424 *defined.*
- 12425 **13.8.3.4.1 Work unit ALC_TAT.3-1**
- 12426 The evaluator ***shall examine*** the development tool documentation provided to determine that
12427 each development tool is well-defined.
- 12428 For example, a well-defined language, compiler or CAD system may be considered to be one that
12429 conforms to a recognised standard, such as the ISO standards. A well-defined language is one that
12430 has a clear and complete description of its syntax, and a detailed description of the semantics of
12431 each construct.
- 12432 At this level, the documentation of development tools used by third party contributors to the TOE
12433 has to be included in the evaluator's examination.
- 12434 ISO/IEC 15408-3 ALC_TAT.3.2C: *The documentation of each development tool shall unambiguously*
12435 *define the meaning of all statements as well as all conventions and directives used in the*
12436 *implementation.*
- 12437 **13.8.3.4.2 Work unit ALC_TAT.3-2**
- 12438 The evaluator ***shall examine*** the documentation of each development tool to determine that it
12439 unambiguously defines the meaning of all statements as well as all conventions and directives used
12440 in the implementation.

- 12441 The development tool documentation (e.g. programming language specifications and user
12442 manuals) should cover all statements used in the implementation representation of the TOE, and
12443 for each such statement should provide a clear and unambiguous definition of the purpose and
12444 effect of that statement. This work may be performed in parallel with the evaluator's examination
12445 of the implementation representation performed during the ADV_IMP sub-activity. The key test the
12446 evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to
12447 be able to understand the implementation representation. The documentation should not assume
12448 (for example) that the reader is an expert in the programming language used.
- 12449 Reference to the use of a documented standard is an acceptable approach to meet this requirement,
12450 provided that the standard is available to the evaluator. Any differences from the standard should
12451 be documented.
- 12452 The critical test is whether the evaluator can understand the TOE source code when performing
12453 source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can
12454 additionally be used in searching for problem areas:
- 12455 a) In the language definition, phrases such as “the effect of this construct is undefined” and
12456 terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.
- 12457 b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a
12458 common source of ambiguity problems.
- 12459 c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is
12460 often poorly defined.
- 12461 Most languages in common use, however well designed, will have some problematic constructs. If
12462 the implementation language is mostly well defined, but some problematic constructs exist, then
12463 an inconclusive verdict should be assigned, pending examination of the source code.
- 12464 The evaluator should verify, during the examination of source code, that any use of the problematic
12465 constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs
12466 precluded by the documented standard are not used.
- 12467 The development tool documentation should define all conventions and directives used in the
12468 implementation.
- 12469 At this level, the documentation of development tools used by third party contributors to the TOE
12470 has to be included in the evaluator's examination.
- 12471 ISO/IEC 15408-3 ALC_TAT.3.3C: *The documentation of each development tool shall unambiguously*
12472 *define the meaning of all implementation-dependent options.*
- 12473 **13.8.3.4.3 Work unit ALC_TAT.3-3**
- 12474 The evaluator ***shall examine*** the development tool documentation to determine that it
12475 unambiguously defines the meaning of all implementation-dependent options.
- 12476 The documentation of software development tools should include definitions of implementation-
12477 dependent options that may affect the meaning of the executable code, and those that are different
12478 from the standard language as documented. Where source code is provided to the evaluator,
12479 information should also be provided on compilation and linking options used.
- 12480 The documentation for hardware design and development tools should describe the use of all
12481 options that affect the output from the tools (e.g. detailed hardware specifications, or actual
12482 hardware).

12483 At this level, the documentation of development tools used by third party contributors to the TOE
12484 has to be included in the evaluator's examination.

12485 **13.8.3.5 Action ALC_TAT.3.2E**

12486 **13.8.3.5.1 Work unit ALC_TAT.3-4**

12487 The evaluator ***shall examine*** aspects of the implementation process to determine that documented
12488 implementation standards have been applied.

12489 This work unit requires the evaluator to analyse the provided implementation representation of
12490 the TOE to determine whether the documented implementation standards have been applied.

12491 The evaluator should verify that constructs excluded by the documented standard are not used.

12492 Additionally, the evaluator should verify the developer's procedures which ensure the application
12493 of the defined standards within the design and implementation process of the TOE. Therefore,
12494 documentary evidence should be supplemented by visiting the development environment. A visit
12495 to the development environment will allow the evaluator to:

12496 a) observe the application of defined standards;

12497 b) examine documentary evidence of application of procedures describing the use of defined
12498 standards;

12499 c) interview development staff to check awareness of the application of defined standards
12500 and procedures.

12501 A development site visit is a useful means of gaining confidence in the procedures being used. Any
12502 decision not to make such a visit should be determined in consultation with the evaluation
12503 authority.

12504 The evaluator compares the provided implementation representation with the description of the
12505 applied implementation standards and verifies their use.

12506 At this level it is required that the complete provided implementation representation of the TSF is
12507 based on implementation standards, including third party contributions. This may require the
12508 evaluator to visit the sites of contributors. The evaluator may consult the configuration list
12509 required by the CM scope (ALC_CMS) to see who has developed which part of the TOE.

12510 Note that parts of the TOE which are not TSF relevant do not need to be examined.

12511 This work unit may be performed in conjunction with the evaluation activities under ADV_IMP.

12512 **14 Class ASE: Security Target evaluation**

12513 **14.1 [PLACE-HOLDER] ST Additional Module Analysis (ASE_AMA)**

12514 If the modularity approach included in ASE_AMA, ADV_MTC, ATE_MTK, ATE_MTT remains in
12515 ISO/IEC 15408-x then work units will be required to cover these.

- 12516 **15** Suggestions for text would be welcomed in response to CD1 review. **If none**
 12517 **are received then this topic will be left to the next revision.**Class ATE:
 12518 **Tests**
- 12519 **15.1 Introduction**
- 12520 **15.1.1.1 The goal of this activity is to determine whether the TOE behaves as described in**
 12521 **the ST and as specified in the evaluation evidence (described in the ADV class). This**
 12522 **determination is achieved through some combination of the developer's own functional**
 12523 **testing of the TSF (Functional tests (ATE_FUN)) and independent testing the TSF by the**
 12524 **evaluator (Objectives**
- 12525 **15.1.1.1** The objective of this sub-activity is to determine whether the developer correctly performed and
 12526 documented the tests in the test documentation and to ensure that testing is structured such as to
 12527 avoid circular arguments about the correctness of the interfaces being tested.
- 12528 **15.1.1.1 Input**
- 12529 **15.1.1.1** The evaluation evidence for this sub-activity is:
- 12530 **15.1.1.1** the ST;
- 12531 **15.1.1.1** the functional specification;
- 12532 **15.1.1.1** the test documentation.
- 12533 **15.1.1.1 Application notes**
- 12534 **15.1.1.1** Although the test procedures may state pre-requisite initial test conditions in terms of ordering
 12535 of tests, they may not provide a rationale for the ordering. An analysis of test ordering, which
 12536 provides this rationale, is an important factor in determining the adequacy of testing, as there is a
 12537 possibility of faults being concealed by the ordering of tests.
- 12538 **15.1.1.1 Action ATE_FUN.2.1E**
- 12539 **15.1.1.1** ISO/IEC 15408-3 ATE_FUN.2.1C *The test documentation shall consist of test plans, expected test*
 12540 *results and actual test results.*
- 12541 **15.1.1.1 Work unit ATE_FUN.2-1**
- 12542 **15.1.1.1** The evaluator *shall check* that the test documentation includes test plans, expected test results
 12543 and actual test results.
- 12544 **15.1.1.1** The evaluator checks that test plans, expected tests results and actual test results are included in
 12545 the test documentation.
- 12546 **15.1.1.1** ISO/IEC 15408-3 ATE_FUN.2.2C *The test plans shall identify the tests to be performed and*
 12547 *describe the scenarios for performing each test. These scenarios shall include any ordering*
 12548 *dependencies on the results of other tests.*
- 12549 **15.1.1.1 Work unit ATE_FUN.2-2**
- 12550 **15.1.1.1** The evaluator *shall examine* the test plan to determine that it describes the scenarios for
 12551 performing each test.
- 12552 **15.1.1.1** The evaluator determines that the test plan provides information about the test configuration
 12553 being used: both on the configuration of the TOE and on any test equipment being used. This
 12554 information should be detailed enough to ensure that the test configuration is reproducible.

- 12555 **15.1.1.1** The evaluator also determines that the test plan provides information about how to
 12556 execute the test: any necessary automated set-up procedures (and whether they require privilege
 12557 to run), inputs to be applied, how these inputs are applied, how output is obtained, any
 12558 automated clean-up procedures (and whether they require privilege to run), etc. This
 12559 information should be detailed enough to ensure that the test is reproducible.
- 12560 **15.1.1.1** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 12561 **15.1.1.1 Work unit ATE_FUN.2-3**
- 12562 **15.1.1.1** The evaluator *shall examine the test plan to determine that the TOE test configuration*
 12563 *is consistent with the ST.*
- 12564 **15.1.1.1** The TOE referred to in the developer's test plan should have the same unique reference
 12565 as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
 12566 introduction.
- 12567 **15.1.1.1** It is possible for the ST to specify more than one configuration for evaluation. The
 12568 evaluator verifies that all test configurations identified in the developer test documentation are
 12569 consistent with the ST. For example, the ST might define configuration options that must be set,
 12570 which could have an impact upon what constitutes the TOE by including or excluding additional
 12571 portions. The evaluator verifies that all such variations of the TOE are considered.
- 12572 **15.1.1.1** The evaluator should consider the security objectives for the operational environment
 12573 described in the ST that may apply to the test environment. There may be some objectives for the
 12574 operational environment that do not apply to the test environment. For example, an objective
 12575 about user clearances may not apply; however, an objective about a single point of connection to
 12576 a network would apply.
- 12577 **15.1.1.1** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 12578 **15.1.1.1** If this work unit is applied to a component TOE that might be used/integrated in a
 12579 composed TOE (see Class ACO: Composition), the following will apply. In the instances that the
 12580 component TOE under evaluation depends on other components in the operational environment
 12581 to support their operation, the developer may wish to consider using the other component(s)
 12582 that will be used in the composed TOE to fulfil the requirements of the operational environment
 12583 as one of the test configurations. This will reduce the amount an additional testing that will be
 12584 required for the composed TOE evaluation.
- 12585 **15.1.1.1 Work unit ATE_FUN.2-4**
- 12586 **15.1.1.1** The evaluator *shall examine the test plans to determine that sufficient instructions are*
 12587 *provided for any ordering dependencies.*
- 12588 **15.1.1.1** Some steps may have to be performed to establish initial conditions. For example, user
 12589 accounts need to be added before they can be deleted. An example of ordering dependencies on
 12590 the results of other tests is the need to perform actions in a test that will result in the generation
 12591 of audit records, before performing a test to consider the searching and sorting of those audit
 12592 records. Another example of an ordering dependency would be where one test case generates a
 12593 file of data to be used as input for another test case.
- 12594 **15.1.1.1** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 12595 **15.1.1.1 ATE_FUN.2.3C** *The expected test results shall show the anticipated outputs*
 12596 *from a successful execution of the tests.*

- 12597 **15.1.1.1 Work unit ATE_FUN.2-5**
- 12598 **15.1.1.1** The evaluator *shall examine the test documentation to determine that all expected*
12599 *tests results are included.*
- 12600 **15.1.1.1** The expected test results are needed to determine whether or not a test has been
12601 successfully performed. Expected test results are sufficient if they are unambiguous and
12602 consistent with expected behaviour given the testing approach.
- 12603 **15.1.1.1** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 12604 **15.1.1.1 ATE_FUN.2.4C** *The actual test results shall be consistent with the expected*
12605 *test results.*
- 12606 **15.1.1.1 Work unit ATE_FUN.2-6**
- 12607 **15.1.1.1** The evaluator *shall check that the actual test results in the test documentation are*
12608 *consistent with the expected test results in the test documentation.*
- 12609 **15.1.1.1** A comparison of the actual and expected test results provided by the developer will
12610 reveal any inconsistencies between the results. It may be that a direct comparison of actual
12611 results cannot be made until some data reduction or synthesis has been first performed. In such
12612 cases, the developer's test documentation should describe the process to reduce or synthesise the
12613 actual data.
- 12614 **15.1.1.1** For example, the developer may need to test the contents of a message buffer after a
12615 network connection has occurred to determine the contents of the buffer. The message buffer will
12616 contain a binary number. This binary number would have to be converted to another form of data
12617 representation in order to make the test more meaningful. The conversion of this binary
12618 representation of data into a higher-level representation will have to be described by the
12619 developer in enough detail to allow an evaluator to perform the conversion process (i.e.
12620 synchronous or asynchronous transmission, number of stop bits, parity, etc.).
- 12621 **15.1.1.1** It should be noted that the description of the process used to reduce or synthesise the
12622 actual data is used by the evaluator not to actually perform the necessary modification but to
12623 assess whether this process is correct. It is up to the developer to transform the expected test
12624 results into a format that allows an easy comparison with the actual test results.
- 12625 **15.1.1.1** The evaluator may wish to employ a sampling strategy when performing this work unit.
- 12626 **15.1.1.1 Work unit ATE_FUN.2-7**
- 12627 **15.1.1.1** The evaluator **shall report the developer testing effort, outlining the testing approach,**
12628 **configuration, depth and results.**
- 12629 **15.1.1.1** The developer testing information recorded in the ETR allows the evaluator to convey
12630 the overall testing approach and effort expended on the testing of the TOE by the developer. The
12631 intent of providing this information is to give a meaningful overview of the developer testing
12632 effort. It is not intended that the information regarding developer testing in the ETR be an exact
12633 reproduction of specific test steps or results of individual tests. The intention is to provide
12634 enough detail to allow other evaluators and evaluation authorities to gain some insight about the
12635 developer's testing approach, amount of testing performed, TOE test configurations, and the
12636 overall results of the developer testing.
- 12637 **15.1.1.1** Information that would typically be found in the ETR section regarding the developer
12638 testing effort is:

- 12639 **15.1.1.1** TOE test configurations. The particular configurations of the TOE that were tested,
12640 including whether any privileged code was required to set up the test or clean up afterwards;
- 12641 **15.1.1.1** testing approach. An account of the overall developer testing strategy employed;
- 12642 **15.1.1.1** testing results. A description of the overall developer testing results.
- 12643 **15.1.1.1** This list is by no means exhaustive and is only intended to provide some context as to
12644 the type of information that should be present in the ETR concerning the developer testing effort.
- 12645 **15.1.1.1 ATE_FUN.2.5C** *The test documentation shall include an analysis of the test*
12646 *procedure ordering dependencies.*
- 12647 **15.1.1.1 Work unit ATE_FUN.2-8**
- 12648 **15.1.1.1** The evaluator *shall examine the analysis of the test procedure ordering dependencies*
12649 to determine that a sufficient justification for the chosen ordering of test cases is given.
- 12650 **15.1.1.1** Usually the evaluator will generate a table of all cases, where the test documentation
12651 requires a certain ordering of the tests and will then examine if sufficient justification is given in
12652 any case, why testing in this ordering is adequate and sufficient.
- 12653 **15.1.1.1** As an example we assume that the TSF provide a random number generator, which
12654 needs to be initialised (for example with an adequate seed) before random numbers of a specified
12655 quality can be generated. In this case the evaluator will consider the following question:
- 12656 **15.1.1.1** Does the test documentation only describe an ordering of tests, where the initialisation
12657 is done before calling the function to generate a random number?
- 12658 **15.1.1.1** In this case the justification needs to show, why the developer expects, that in the
12659 intended environment of the TOE the random number function will not be called without
12660 initialisation of the random number generator.
- 12661 **15.1.1.1** If for example the user guidance documentation includes a clear instruction that the
12662 random number generator needs to be initialised adequately before being called, this may be
12663 considered adequate as a justification. (note that the question if it can be plausibly assumed that
12664 users will follow such instruction is covered by the evaluation activities for the classes ASE and
12665 AGD and needs not to be re-examined here.)
- 12666 **15.1.1.1** If, on the other hand, the TOE provides an authentication protocol, which implicitly uses
12667 random numbers provided by the random number generator, and an attacker can therefore "call"
12668 the random number generator implicitly by simply trying to authenticate himself, and if neither
12669 the TOE nor the environment prevent an attacker from doing this even before the random
12670 number generator is initialised, a test case needs to show, what happens in such situation.
- 12671 **15.1.1.1** If, for example, instead of returning a "bad" random number, the random number
12672 function would return an error, when called without proper initialisation, it would be much
12673 better to include a test showing this secure behaviour instead of trying to justify why the
12674 functions are only tested in the usual order.
- 12675 **15.1.1.1** Note: Of course even without ATE_FUN.2 an evaluator would be expected to look for
12676 potential vulnerabilities like the one described above. However, ATE_FUN.2.5C adds assurance by
12677 requiring the developer to give a systematic justification, why their **chosen order of test cases**
12678 doesn't hide such potential failures of security functions.
- 12679 **15.1.1.1** Independent testing (ATE_IND)). At the lowest level of assurance, there is no
12680 requirement for developer involvement, so the only testing is conducted by the evaluator, using the
12681 limited available information about the TOE. Additional assurance is gained as the developer

12682 becomes increasingly involved both in testing and in providing additional information about the
12683 TOE, and as the evaluator increases the independent testing activities.

12684 **15.2 Application notes**

12685 Testing of the TSF is conducted by the evaluator and, in most cases, by the developer. The
12686 evaluator's testing efforts consist not only of creating and running original tests, but also of
12687 assessing the adequacy of the developer's tests and re-running a subset of them.

12688 The evaluator analyses the developer's tests to determine the extent to which they are sufficient to
12689 demonstrate that TSFI (see Functional specification (ADV_FSP)) perform as specified, and to
12690 understand the developer's approach to testing. Similarly, the evaluator analyses the developer's
12691 tests to determine the extent to which they are sufficient to demonstrate the internal behaviour
12692 and properties of the TSF.

12693 The evaluator also executes a subset of the developer's tests as documented to gain confidence in
12694 the developer's test results: the evaluator will use the results of this analysis as an input to
12695 independently testing a subset of the TSF. With respect to this subset, the evaluator takes a testing
12696 approach that is different from that of the developer, particularly if the developer's tests have
12697 shortcomings.

12698 **15.2.1.1 To determine the adequacy of developer's test documentation or to create new**
12699 **tests, the evaluator needs to understand the desired expected behaviour of the TSF, both**
12700 **internally and as seen at the TSFI, in the context of the SFRs it is to satisfy. The evaluator**
12701 **may choose to divide the TSF and TSFI into subsets according to functional areas of the ST**
12702 **(audit subsystem, audit-related TSFI, authentication module, authentication-related TSFI,**
12703 **etc.) if they were not already divided in the ST, and focus on one subset of the TSF and TSFI**
12704 **at a time, examining the ST requirement and the relevant parts of the development and**
12705 **guidance documentation to gain an understanding of the way the TOE is expected to**
12706 **behave. This reliance upon the development documentation underscores the need for the**
12707 **dependencies on ADV by Coverage (ATE_COV) and Objectives**

12708 **15.2.1.1** The objective of this sub-activity is to determine whether the developer has tested all of the TSFIs
12709 exhaustively, and that the developer's test coverage evidence shows correspondence between the
12710 tests identified in the test documentation and the TSFIs described in the functional specification.

12711 **15.2.1.1** A particular objective of this component is to confirm that all parameters of all of the TSFIs have
12712 been tested.

12713 **15.2.1.1 Input**

12714 **15.2.1.1** The evaluation evidence for this sub-activity is:

12715 **15.2.1.1** the ST;

12716 **15.2.1.1** the functional specification;

12717 **15.2.1.1** the test documentation;

12718 **15.2.1.1** the test coverage analysis.

12719 **15.2.1.1 Action ATE_COV.3.1E**

12720 **15.2.1.1** ISO/IEC 15408-3 ATE_COV.3.1C: *The analysis of the test coverage shall demonstrate the*
12721 *correspondence between the tests in the test documentation and the TSFIs in the functional*
12722 *specification.*

12723	15.2.1.1 Work unit ATE_COV.3-1
12724	15.2.1.1 The evaluator <i>shall examine the test coverage analysis to determine that the</i>
12725	
12726	
12727	15.2.1.1 A simple cross-table may be sufficient to show test correspondence. The identification
12728	
12729	15.2.1.1 The evaluator is reminded that this does not imply that all tests in the test
12730	
12731	15.2.1.1 Work unit ATE_COV.3-2
12732	15.2.1.1 The evaluator <i>shall examine the test plan to determine that the testing approach for</i>
12733	
12734	15.2.1.1 Guidance on this work unit can be found in:
12735	15.2.1.1 15.2.1 Understanding the expected behaviour of the TOE
12736	15.2.1.1 15.2.2 [Testing vs. alternate approaches to verify the expected behaviour of
12737	
12738	15.2.1.1 Work unit ATE_COV.3-3
12739	15.2.1.1 The evaluator <i>shall examine the test procedures to determine that the test</i>
12740	
12741	15.2.1.1 Guidance on this work units, as it pertains to the functional specification, can be found
12742	
12743	15.2.1.1 15.2.3 Verifying the adequacy of tests
12744	15.2.1.1 ISO/IEC 15408-3 ATE_COV.3.2C <i>The analysis of the test coverage shall demonstrate</i>
12745	
12746	15.2.1.1 Work unit ATE_COV.3-4
12747	15.2.1.1 The evaluator <i>shall examine the test coverage analysis to determine that the</i>
12748	
12749	
12750	15.2.1.1 All TSFIs that are described in the functional specification have to be present in the test
12751	
12752	
12753	
12754	15.2.1.1 The evaluator is reminded that this does not imply that all tests in the test
12755	
12756	15.2.1.1 Work unit ATE_COV.3-5
12757	15.2.1.1 The evaluator <i>shall examine the test coverage analysis to determine that the</i>
12758	
12759	

- 12760 **15.2.1.1** This means that the evaluator examines whether all aspects of purpose, method of use,
 12761 parameters, parameter descriptions, actions and error messages for all TSFIs present in the
 12762 functional specification are covered by the tests. Note that the level of detail present in the
 12763 functional specification depends on the component of ADV_FSP chosen in the ST of the TOE.
- 12764 **15.2.1.1** The evaluator may conclude that the higher level descriptions in the functional
 12765 specification, like purpose or method of use, are implicitly covered, if coverage of lower level
 12766 descriptions like parameters, parameter descriptions, actions and error messages are covered.
 12767 Therefore in general it will only be necessary to confirm coverage on these lower levels.
- 12768 **15.2.1.1** The evaluator is reminded that (for example) coverage of all parameters does not
 12769 necessarily mean coverage of every possible value a parameter may allow. However every value
 12770 for which a distinct qualitative behaviour of the TOE is expected, needs to be covered.
- 12771 **15.2.1.1** As an example: If one of the parameters of a function call is a two byte value, which
 12772 specifies the length of further parameters, only some typical values need to be tested. However
 12773 the evaluator will make sure that some specific cases (like the value zero or the maximal value)
 12774 will be covered.
- 12775 **15.2.1.1** If the evaluator sees that a potential attacker might be able to invoke a TSFI with
 12776 inconsistent parameter values (e. g. if one parameter specifies the length of a second parameter
 12777 and it is possible to make the second parameter actually longer than the chosen value for the first
 12778 parameter suggests) and this case is not covered by the developer's testing, the evaluator may
 12779 decide either to test this during their **activities in AVA_VAN** or to **require the developer to provide**
 12780 coverage also for this case.
- 12781 **15.2.1.1** Similar considerations as for parameters hold for error messages specified in the
 12782 functional specification: Each error message, which belongs to a qualitatively distinct error case,
 12783 needs to be covered by testing. Note, that there may be exceptions, for example error messages
 12784 for errors, which cannot be provoked during testing. For such error messages other ways of
 12785 coverage need to be found as discussed in 15.2.2, "**Testing vs. alternate approaches to verify the**
 12786 **expected behaviour of functionality**".
- 12787 **15.2.1.1** Note that also the developer is allowed to use such alternative approaches to testing (e.
 12788 g. checking something in the source code) in the **coverage table**. **Of course the evaluator has to**
 12789 **examine in this case, if this use of an alternative approach is acceptable** (usually only in cases
 12790 where testing is practically impossible).
- 12791 **15.2.1.1** Depth (ATE_DPT).
- 12792 ISO/IEC 15408 has separated coverage and depth from functional tests to increase the flexibility
 12793 when applying the components of the families. However, the requirements of the families are
 12794 intended to be applied together to confirm that the TSF operates according to its specification. This
 12795 tight coupling of families has led to some duplication of evaluator work units across sub-activities.
 12796 These application notes are used to minimise duplication of text between sub-activities.
- 12797 **15.2.2 Understanding the expected behaviour of the TOE**
- 12798 Before the adequacy of test documentation can be accurately evaluated, or before new tests can be
 12799 created, the evaluator has to understand the desired expected behaviour of a security function in
 12800 the context of the requirements it is to satisfy.
- 12801 As mentioned earlier, the evaluator may choose to subset the TSF and TSFI according to SFRs
 12802 (audit, authentication, etc.) in the ST and focus on one subset at a time. The evaluator examines
 12803 each ST requirement and the relevant parts of the functional specification and guidance
 12804 documentation to gain an understanding of the way the related TSFI is expected to behave.
 12805 Similarly, the evaluator examines the relevant parts of the TOE design and security architecture

12806 documentation to gain an understanding of the way the related modules or subsystems of the TSF
12807 are expected to behave.

12808 With an understanding of the expected behaviour, the evaluator examines the test plan to gain an
12809 understanding of the testing approach. In most cases, the testing approach will entail a TSFI being
12810 stimulated and its responses observed. Externally-visible functionality can be tested directly;
12811 however, in cases where functionality is not visible external to the TOE (for example, testing the
12812 residual information protection functionality), other means will need to be employed.

12813 **15.2.3 Testing vs. alternate approaches to verify the expected behaviour of functionality**

12814 In cases where it is impractical or inadequate to test specific functionality (where it provides no
12815 externally-visible TSFI), the test plan should identify the alternate approach to verify expected
12816 behaviour. It is the evaluator's responsibility to determine the suitability of the alternate approach.
12817 However, the following should be considered when assessing the suitability of alternate
12818 approaches:

12819 a) an analysis of the implementation representation to determine that the required
12820 behaviour should be exhibited by the TOE is an acceptable alternate approach. This could
12821 mean a code inspection for a software TOE or perhaps a chip mask inspection for a
12822 hardware TOE.

12823 b) it is acceptable to use evidence of developer integration or module testing, even if the
12824 claimed assurance requirements do not include availability of lower level descriptions of
12825 the TOE modules (e.g. Evaluation of sub-activity (ADV_TDS.3)) or implementation
12826 (Implementation representation (ADV_IMP)). If evidence of developer integration or
12827 module testing is used in verifying the expected behaviour of a security functionality,
12828 care should be given to confirm that the testing evidence reflects the current
12829 implementation of the TOE. If the subsystems or modules have been changed since
12830 testing occurred, evidence that the changes were tracked and addressed by analysis or
12831 further testing will usually be required.

12832 It should be emphasised that supplementing the testing effort with alternate approaches should
12833 only be undertaken when both the developer and evaluator determine that there exists no other
12834 practical means to test the expected behaviour.

12835 **15.2.4 Verifying the adequacy of tests**

12836 Test pre-requisites are necessary to establish the required initial conditions for the test. They may
12837 be expressed in terms of parameters that must be set or in terms of test ordering in cases where
12838 the completion of one test establishes the necessary pre-requisites for another test. The evaluator
12839 must determine that the pre-requisites are complete and appropriate in that they will not bias the
12840 observed test results towards the expected test results.

12841 The test steps and expected results specify the actions and parameters to be applied to the TSFI as
12842 well as how the expected results should be verified and what they are. The evaluator must
12843 determine that the test steps and expected results are consistent with the descriptions of the TSFI
12844 in the functional specification. This means that each characteristic of the TSFI behaviour explicitly
12845 described in the functional specification should have tests and expected results to verify that
12846 behaviour.

12847 The overall aim of this testing activity is to determine that each subsystem, module, and TSFI has
12848 been sufficiently tested against the behavioural claims in the functional specification, TOE design,
12849 and architecture description. At the higher assurance levels, testing also includes bounds testing
12850 and negative testing. The test procedures will provide insight as to how the TSFIs, modules, and
12851 subsystems have been exercised by the developer during testing. The evaluator uses this
12852 information when developing additional tests to independently test the TSF.

12853 **15.3 Coverage (ATE_COV)**

12854 **15.3.1 Evaluation of sub-activity (ATE_COV.1)**

12855 **15.3.1.1 Objectives**

12856 The objective of this sub-activity is to determine whether the developer has tested the TSFIs, and
12857 that the developer's test coverage evidence shows correspondence between the tests identified in
12858 the test documentation and the TSFIs described in the functional specification.

12859 **15.3.1.2 Input**

12860 The evaluation evidence for this sub-activity is:

- 12861 a) the ST;
- 12862 b) the functional specification;
- 12863 c) the test documentation;
- 12864 d) the test coverage evidence.

12865 **15.3.1.3 Application notes**

12866 The coverage analysis provided by the developer is required to show the correspondence between
12867 the tests provided as evaluation evidence and the functional specification. However, the coverage
12868 analysis need not demonstrate that all TSFI have been tested, or that all externally-visible
12869 interfaces to the TOE have been tested. Such shortcomings are considered by the evaluator during
12870 the independent testing (Evaluation of sub-activity (ATE_IND.2)) sub-activity.

12871 **15.3.1.4 Action ATE_COV.1.1E**

12872 ISO/IEC 15408-3 ATE_COV.1.1C: *The evidence of the test coverage shall show the correspondence*
12873 *between the tests in the test documentation and the TSFIs in the functional specification.*

12874 **15.3.1.4.1 Work unit ATE_COV.1-1**

12875 The evaluator ***shall examine*** the test coverage evidence to determine that the correspondence
12876 between the tests identified in the test documentation and the TSFIs described in the functional
12877 specification is accurate.

12878 Correspondence may take the form of a table or matrix. The coverage evidence required for this
12879 component will reveal the extent of coverage, rather than to show complete coverage. In cases
12880 where coverage is shown to be poor the evaluator should increase the level of independent testing
12881 to compensate.

12882 **15.3.2 Evaluation of sub-activity (ATE_COV.2)**

12883 **15.3.2.1 Objectives**

12884 The objective of this sub-activity is to determine whether the developer has tested all of the TSFIs,
12885 and that the developer's test coverage evidence shows correspondence between the tests
12886 identified in the test documentation and the TSFIs described in the functional specification.

12887 **15.3.2.2 Input**

- 12888 a) the ST;

12889 b) the functional specification;

12890 c) the test documentation;

12891 d) the test coverage analysis.

12892 **15.3.2.3 Action ATE_COV.2.1E**

12893 ISO/IEC 15408-3 ATE_COV.2.1C: *The analysis of the test coverage shall demonstrate the*
12894 *correspondence between the tests in the test documentation and the TSFIs in the functional*
12895 *specification.*

12896 **15.3.2.3.1 Work unit ATE_COV.2-1**

12897 The evaluator **shall examine** the test coverage analysis to determine that the correspondence
12898 between the tests in the test documentation and the interfaces in the functional specification is
12899 accurate.

12900 A simple cross-table may be sufficient to show test correspondence. The identification of the tests
12901 and the interfaces presented in the test coverage analysis has to be unambiguous.

12902 The evaluator is reminded that this does not imply that all tests in the test documentation must
12903 map to interfaces in the functional specification.

12904 **15.3.2.3.2 Work unit ATE_COV.2-2**

12905 The evaluator **shall examine** the test plan to determine that the testing approach for each interface
12906 demonstrates the expected behaviour of that interface.

12907 Guidance on this work unit can be found in:

12908 a) 15.2.2, Understanding the expected behaviour of the TOE

12909 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

12910 **15.3.2.3.3 Work unit ATE_COV.2-3**

12911 The evaluator **shall examine** the test procedures to determine that the test prerequisites, test
12912 steps and expected result(s) adequately test each interface.

12913 Guidance on this work units, as it pertains to the functional specification, can be found in:

12914 a) 15.2.4, Verifying the adequacy of tests

12915 ISO/IEC 15408-3 ATE_COV.2.2C: *The analysis of the test coverage shall demonstrate that all TSFIs in*
12916 *the functional specification have been tested.*

12917 **15.3.2.3.4 Work unit ATE_COV.2-4**

12918 The evaluator **shall examine** the test coverage analysis to determine that the correspondence
12919 between the interfaces in the functional specification and the tests in the test documentation is
12920 complete.

12921 All TSFIs that are described in the functional specification have to be present in the test coverage
12922 analysis and mapped to tests in order for completeness to be claimed, although exhaustive
12923 specification testing of interfaces is not required. Incomplete coverage would be evident if an
12924 interface was identified in the functional specification and no test was mapped to it.

12925 The evaluator is reminded that this does not imply that all tests in the test documentation must
12926 map to interfaces in the functional specification.

12927 **15.3.3 Evaluation of sub-activity (ATE_COV.3)**

12928 **15.3.3.1 Objectives**

12929 The objective of this sub-activity is to determine whether the developer has tested all of the TSFIs
12930 exhaustively, and that the developer's test coverage evidence shows correspondence between the
12931 tests identified in the test documentation and the TSFIs described in the functional specification.

12932 A particular objective of this component is to confirm that all parameters of all of the TSFIs have
12933 been tested.

12934 **15.3.3.2 Input**

12935 The evaluation evidence for this sub-activity is:

- 12936 e) the ST;
- 12937 f) the functional specification;
- 12938 g) the test documentation;
- 12939 h) the test coverage analysis.

12940 **15.3.3.3 Action ATE_COV.3.1E**

12941 ISO/IEC 15408-3 ATE_COV.3.1C: *The analysis of the test coverage shall demonstrate the*
12942 *correspondence between the tests in the test documentation and the TSFIs in the functional*
12943 *specification.*

12944 **15.3.3.3.1 Work unit ATE_COV.3-1**

12945 The evaluator ***shall examine*** the test coverage analysis to determine that the correspondence
12946 between the tests in the test documentation and the interfaces in the functional specification is
12947 accurate.

12948 A simple cross-table may be sufficient to show test correspondence. The identification of the tests
12949 and the interfaces presented in the test coverage analysis has to be unambiguous.

12950 The evaluator is reminded that this does not imply that all tests in the test documentation must
12951 map to interfaces in the functional specification.

12952 **15.3.3.3.2 Work unit ATE_COV.3-2**

12953 The evaluator ***shall examine*** the test plan to determine that the testing approach for each interface
12954 demonstrates the expected behaviour of that interface.

12955 Guidance on this work unit can be found in:

- 12956 c) 15.2.1 Understanding the expected behaviour of the TOE
- 12957 d) 15.2.2 [Testing vs. alternate approaches to verify the expected
12958 behaviour of functionality]

12959 **15.3.3.3.3 Work unit ATE_COV.3-3**

12960 The evaluator **shall examine** the test procedures to determine that the test prerequisites, test
12961 steps and expected result(s) adequately test each interface.

12962 Guidance on this work units, as it pertains to the functional specification, can be found in:

12963 b) 15.2.3 Verifying the adequacy of tests

12964 ISO/IEC 15408-3 ATE_COV.3.2C *The analysis of the test coverage shall demonstrate that all TSFIs in*
12965 *the functional specification have been completely tested.*

12966 **15.3.3.3.4 Work unit ATE_COV.3-4**

12967 The evaluator **shall examine** the test coverage analysis to determine that the correspondence
12968 between the interfaces in the functional specification and the tests in the test documentation is
12969 complete.

12970 All TSFIs that are described in the functional specification have to be present in the test coverage
12971 analysis and mapped to tests in order for completeness to be claimed. Exhaustive specification
12972 testing of interfaces is required for this mapping. Incomplete coverage would be evident if an
12973 interface was identified in the functional specification and no test was mapped to it.

12974 The evaluator is reminded that this does not imply that all tests in the test documentation must
12975 map to interfaces in the functional specification.

12976 **15.3.3.3.5 Work unit ATE_COV.3-5**

12977 The evaluator **shall examine** the test coverage analysis to determine that the correspondence
12978 between the interfaces in the functional specification and the tests in the test documentation shows
12979 that all TSFIs were tested completely.

12980 This means that the evaluator examines whether all aspects of purpose, method of use, parameters,
12981 parameter descriptions, actions and error messages for all TSFIs present in the functional
12982 specification are covered by the tests. Note that the level of detail present in the functional
12983 specification depends on the component of ADV_FSP chosen in the ST of the TOE.

12984 The evaluator may conclude that the higher level descriptions in the functional specification, like
12985 purpose or method of use, are implicitly covered, if coverage of lower level descriptions like
12986 parameters, parameter descriptions, actions and error messages are covered. Therefore in general
12987 it will only be necessary to confirm coverage on these lower levels.

12988 The evaluator is reminded that (for example) coverage of all parameters does not necessarily mean
12989 coverage of every possible value a parameter may allow. However every value for which a distinct
12990 qualitative behaviour of the TOE is expected, needs to be covered.

12991 As an example: If one of the parameters of a function call is a two byte value, which specifies the
12992 length of further parameters, only some typical values need to be tested. However the evaluator
12993 will make sure that some specific cases (like the value zero or the maximal value) will be covered.

12994 If the evaluator sees that a potential attacker might be able to invoke a TSFI with inconsistent
12995 parameter values (e. g. if one parameter specifies the length of a second parameter and it is
12996 possible to make the second parameter actually longer than the chosen value for the first
12997 parameter suggests) and this case is not covered by the developer's testing, the evaluator may
12998 decide either to test this during their activities in AVA_VAN or to require the developer to provide
12999 coverage also for this case.

13000 Similar considerations as for parameters hold for error messages specified in the functional
13001 specification: Each error message, which belongs to a qualitatively distinct error case, needs to be

13002 covered by testing. Note, that there may be exceptions, for example error messages for errors,
 13003 which cannot be provoked during testing. For such error messages other ways of coverage need to
 13004 be found as discussed in 15.2.2, "Testing vs. alternate approaches to verify the expected behaviour
 13005 of functionality".

13006 Note that also the developer is allowed to use such alternative approaches to testing (e. g. checking
 13007 something in the source code) in the coverage table. Of course the evaluator has to examine in this
 13008 case, if this use of an alternative approach is acceptable (usually only in cases where testing is
 13009 practically impossible).

13010 **15.4 Depth (ATE_DPT)**

13011 **15.4.1 Evaluation of sub-activity (ATE_DPT.1)**

13012 **15.4.1.1 Objectives**

13013 The objective of this sub-activity is to determine whether the developer has tested the TSF
 13014 subsystems against the TOE design and the security architecture description.

13015 **15.4.1.2 Input**

- 13016 a) the ST;
- 13017 b) the functional specification;
- 13018 c) the TOE design;
- 13019 d) the security architecture description;
- 13020 e) the test documentation;
- 13021 f) the depth of testing analysis.

13022 **15.4.1.3 Action ATE_DPT.1.1E**

13023 ISO/IEC 15408-3 ATE_DPT.1.1C: *The analysis of the depth of testing shall demonstrate the*
 13024 *correspondence between the tests in the test documentation and the TSF subsystems in the TOE*
 13025 *design.*

13026 **15.4.1.3.1 Work unit ATE_DPT.1-1**

13027 The evaluator ***shall examine*** the depth of testing analysis to determine that the descriptions of the
 13028 behaviour of TSF subsystems and of their interactions is included within the test documentation.

13029 This work unit verifies the content of the correspondence between the tests and the descriptions in
 13030 the TOE design. In cases where the description of the TSF's architectural soundness (in Security
 13031 Architecture (ADV_ARC)) cites specific mechanisms, this work unit also verifies the
 13032 correspondence between the tests and the descriptions of the behaviour of such mechanisms.

13033 A simple cross-table may be sufficient to show test correspondence. The identification of the tests
 13034 and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

13035 **15.4.1.4 When Evaluation of sub-activity (ATE_DPT.1) is combined with a component of** 13036 **Objectives**

13037 **15.4.1.4** The objectives of this sub-activity are to determine whether the formal security policy model of
 13038 the TSF clearly and consistently describes the rules and characteristics of the security policies

13039	and whether this description corresponds with the description of security functions in the
13040	functional specification.
13041	15.4.1.4 Input
13042	15.4.1.4 The evaluation evidence for this sub-activity is:
13043	15.4.1.4 the ST;
13044	15.4.1.4 the functional specification;
13045	15.4.1.4 formal security policy model (ADV_SPM.1.1D);
13046	15.4.1.4 formal proof of correspondence between the model and any formal functional
13047	specification (ADV_SPM.1.3D);
13048	15.4.1.4 demonstration of correspondence between the model and the functional specification
13049	(ADV_SPM.1.4D).
13050	15.4.1.4 Application notes
13051	15.4.1.4 This activity applies to cases where the developer has provided a formal security policy
13052	model of the TOE.
13053	15.4.1.4 A formal TOE security policy model is a representation of the rules (synonymously
13054	termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
13055	terms. Their formal counterparts are called security properties and security features,
13056	respectively. The representation includes but is not limited to algebraic specifications, finite state
13057	machines and logic formalisms strong enough to formally infer the properties from the features.
13058	The formal TSP model is accompanied by an informal interpretation explaining how the rules and
13059	characteristics are mapped to the respective properties and features.
13060	15.4.1.4 The creation of a formal security policy model helps to identify and eliminate
13061	ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
13062	has been built, the formal model serves the evaluation effort by contributing to the evaluator's
13063	judgement of how well the developer has understood the security functionality being
13064	implemented and whether there are inconsistencies between the security requirements and the
13065	TOE design. The confidence in the model is accompanied by a proof that it contains no
13066	inconsistencies.
13067	15.4.1.4 A formal security model is a precise formal presentation of the important aspects of
13068	security and their relationship to the behaviour of the TOE; it identifies the set of rules
13069	(principles) that defines the TOE security policy and the set of practises (characteristics) that
13070	regulates how the TSF manages, protects, and otherwise controls the system resources. The
13071	model includes the set of restrictions and properties that specify how information and computing
13072	resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
13073	engineering arguments showing that these restrictions and properties play a key role in the
13074	enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
13075	as well as ancillary text to explain the model and to provide it with context. The security
13076	behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
13077	with the rest of the TOE and with its operational environment), as well as its internal behaviour.
13078	15.4.1.4 The Security Policy Model of the TOE is informally abstracted from its realisation by
13079	considering the proposed security requirements of the ST. The informal abstraction is taken to be
13080	successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
13081	formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
13082	are always prone to fallacies; especially if relationships among subjects, objects and operations
13083	get more and more involved. In order to minimise the risk of insecure state arrivals the rules and

- 13084 characteristics of the security policy model are mapped to respective properties and features
 13085 within some formal system, whose rigour and strength can afterwards be used to obtain the
 13086 security properties by means of theorems and formal proof.
- 13087 **15.4.1.4** While the term “formal security policy model” is used in academic circles, the CC's
 13088 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
 13089 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 13090 of SFRs being claimed.
- 13091 **15.4.1.4** The term security policy has traditionally been associated with only access control
 13092 policies, whether label-based (mandatory access control) or user-based (discretionary access
 13093 control). However, a security policy is not limited to access control; there are also audit policies,
 13094 identification policies, authentication policies, encryption policies, management policies, and any
 13095 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 13096 contains an assignment for identifying these policies that are formally modelled.
- 13097 **15.4.1.4** It is recognized that not all policies can be formally modelled for all TOEs. This is
 13098 because either a given policy can not be formally modelled in the otherwise well suited
 13099 framework, or because the nature of the TOE renders impossible the modelling of policies that
 13100 would otherwise be possible to model.
- 13101 **15.4.1.4 Action ADV_SPM.1.1E**
- 13102 **15.4.1.4 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 13103 *text as required, and identify the security policies of the TSF that are modelled.*
- 13104 **15.4.1.4 Work unit ADV_SPM.1-1**
- 13105 **15.4.1.4** The evaluator *shall examine the TOE security policy model to determine that it is*
 13106 *written in a formal style.*
- 13107 **15.4.1.4** The evaluator identifies the formal framework upon which the TOE security policy
 13108 model is based and ensures that it is founded on well established mathematical concepts. **They**
 13109 **also identify the security properties and features addressed in the application notes and ensure**
 13110 **the formalization of at least one security policy.**
- 13111 **15.4.1.4** For guidance on formal methods refer to ISO/IEC 15408-3
- 13112 **15.4.1.4 Work unit ADV_SPM.1-2**
- 13113 **15.4.1.4** The evaluator *shall examine the TOE security policy model to determine that it*
 13114 *contains all necessary informal explanatory text.*
- 13115 **15.4.1.4** Supporting narrative descriptions are necessary for all parts of the model (for example,
 13116 to make clear the meaning of any formal notation and how they are used) including the security
 13117 properties and features.
- 13118 **15.4.1.4 Work unit ADV_SPM.1-3**
- 13119 **15.4.1.4** The evaluator *shall examine the TOE security policy model to determine that all*
 13120 *security policies of the TSF are identified that are modelled.*
- 13121 **15.4.1.4** The evaluator determines whether the SPM identifies the security policies for which a
 13122 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 13123 of the modelled policies.
- 13124 **15.4.1.4** The evaluator determines whether the list of security policies identified by the SPM is
 13125 consistent with the assignment of ADV_SPM.1.1D in the ST.

- 13126 **15.4.1.4** The evaluator determines whether for each security policy identified by the SPM a
13127 model is in fact provided.
- 13128 **15.4.1.4 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
13129 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
13130 *not secure.*
- 13131 **15.4.1.4 Work unit ADV_SPM.1-4**
- 13132 **15.4.1.4** The evaluator ***shall examine the principles and characteristics of the security policies***
13133 to determine that the modelled security behaviour of the TOE is clearly articulated.
- 13134 **15.4.1.4** The security policies are expressed in terms of security principles (rules) which are
13135 modelled by security properties and define the secure state of the TOE. For example, a model
13136 based on state transitions could describe the security policies in terms of principles of its states,
13137 identify its initial state, and define what it means to be a secure state.
- 13138 **15.4.1.4** The evaluator determines that the security policies are reflected within their formal
13139 counterparts of the TSP model.
- 13140 **15.4.1.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
13141 portions of TOE security functionality managing, protecting, and otherwise controlling the system
13142 resources including attributes and conditions of the TOE) which are modelled by security
13143 features. For example, a model based on state transitions could describe the characteristics as
13144 possible actions in each secure state in a level of detail sufficient to decide into which state the
13145 TOE will be transformed by that action.
- 13146 **15.4.1.4** Together the security principles and characteristics describe the entire security posture
13147 of the TOE.
- 13148 **15.4.1.4** In the context of a formal TOE security policy model the security behaviour is
13149 considered to be clearly articulated only if an adequate mapping from principles and
13150 characteristics to their respective formal counterparts properties and features has been given.
13151 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
13152 detailed enough to allow for correct identification of all security objectives and the relation to the
13153 security environment.
- 13154 **15.4.1.4** The above condition for clear articulation is necessary but not sufficient. An informal
13155 interpretation of all formal concepts (including attributes, predicates and variables, if available)
13156 must be provided in order to make clear their intended meaning.
- 13157 **15.4.1.4 Work unit ADV_SPM.1-5**
- 13158 **15.4.1.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
13159 ***it formally proves that the security features enforce the security properties.***
- 13160 **15.4.1.4** To determine the enforcement, the evaluator considers the security properties and the
13161 security features and verifies that the arguments used in the proof are valid. The proof of
13162 correspondence between the security properties and the security features shall be formal.
- 13163 **15.4.1.4** The validity of the security properties shall mean that the TOE is in a secure state. By
13164 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
13165 state.
- 13166 **15.4.1.4 Work unit ADV_SPM.1-6**
- 13167 **15.4.1.4** The evaluator ***shall examine the TOE security policy model rationale to determine that***
13168 ***it proves the internal consistency of the TOE security policy model.***

- 13169 **15.4.1.4** The proof shall show the absence of contradictions within the TOE security policy
 13170 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 13171 used in the proof are valid.
- 13172 **15.4.1.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
 13173 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 13174 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 13175 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 13176 security policy model that prove the internal consistency by means of a combination with generic
 13177 arguments of the formal framework.
- 13178 **15.4.1.4 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 13179 *specification shall be at the correct level of formality.*
- 13180 **15.4.1.4 Work unit ADV_SPM.1-7**
- 13181 **15.4.1.4** The evaluator *shall examine the correspondence between the model and the functional*
 13182 *specification* to determine that a semiformal demonstration of correspondence between the
 13183 model and any semiformal functional specification is provided.
- 13184 **15.4.1.4** This work unit is only applicable to a semiformal presentation of the functional
 13185 specification, which is required by ADV_FSP.5.2C.
- 13186 **15.4.1.4** A semiformal correspondence is one that results from a structured approach with a
 13187 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 13188 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 13189 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 13190 **15.4.1.4** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 13191 **methods**'.
- 13192 **15.4.1.4 Work unit ADV_SPM.1-8**
- 13193 **15.4.1.4** The evaluator *shall examine the correspondence between the model and the functional*
 13194 *specification* to determine that a formal proof of correspondence between the model and any
 13195 formal functional specification is provided.
- 13196 **15.4.1.4** This work unit is only applicable to a formal presentation of the functional specification,
 13197 which is required by ADV_FSP.6.2D.
- 13198 **15.4.1.4** There should be a formal proof of correspondence between the model and any formal
 13199 functional specification.
- 13200 **15.4.1.4** The formal proof of correspondence removes all subjective interpretations of its terms
 13201 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 13202 formal notation and uses rules that support logical reasoning. The security features within the
 13203 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 13204 language and shown to be satisfied by the formal specification.
- 13205 **15.4.1.4** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 13206 **15.4.1.4 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 13207 *specification is consistent and complete with respect to the model.*

- 13208 **15.4.1.4 Work unit ADV_SPM.1-9**
- 13209 **15.4.1.4** The evaluator *shall examine the correspondence to determine that the behaviour*
 13210 at the TSF interfaces (as articulated in the functional specification) is complete with respect to
 13211 the behaviour modelled by the security features.
- 13212 **15.4.1.4** The term “correspondence” here means both the formal proof of correspondence
 13213 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 13214 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 13215 **15.4.1.4** In determining completeness of the correspondence, the evaluator considers the
 13216 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 13217 features of the TSP model. The demonstration should show that all characteristics belonging to
 13218 policies that are required to be modelled have an associated feature description in the TOE
 13219 security policy model, and that each feature of the TSP model does occur in the mapping.
- 13220 **15.4.1.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
 13221 developer’s side (also confer the application notes above).
- 13222 **15.4.1.4 Work unit ADV_SPM.1-10**
- 13223 **15.4.1.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 13224 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 13225 behaviour modelled by the security features.
- 13226 **15.4.1.4** The term “correspondence” here means both the formal proof of correspondence
 13227 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 13228 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 13229 **15.4.1.4** The meaning of consistency reflects the conventional understanding in contrast to the
 13230 internal consistency concept of work unit ADV_SPM.1-6.
- 13231 **15.4.1.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 13232 security features established in the preceding work unit and verifies that the correspondence
 13233 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 13234 behaviour.
- 13235 **15.4.1.4** For example, if TSFI behaviour dealt with access management on the granularity of
 13236 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 13237 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 13238 management for groups of users, then a TSP model describing the security behaviour of the TOE
 13239 in terms of individual users would also not be consistent.
- 13240 **15.4.1.4** As another example, if remote untrusted users had to pass more stringent
 13241 authentication procedures than administrators whose only point of access were within a
 13242 physically-protected area, then this difference in authentication procedures had to be reflected in
 13243 the security features.
- 13244 **15.4.1.4** TOE design (ADV_TDS), which includes descriptions at the module level (e.g. Evaluation
 13245 of sub-activity (ADV_TDS.3)), the level of detail needed to map the test cases to the behaviour of the
 13246 subsystems may require information from the module description to be used. This is because
 13247 Evaluation of sub-activity (ADV_TDS.3) allows the description of details to be shifted from the
 13248 subsystem level to the module level, or even to omit the subsystems altogether.
- 13249 In any case, the required level of detail in the provided reference to the tested behaviour can be
 13250 defined as “the level of detail required for the description of subsystem behaviour as defined by
 13251 Evaluation of sub-activity (ADV_TDS.2) (in particular work unit ADV_TDS.2-4)”. It states that a
 13252 detailed description of the behaviour typically discusses how the functionality is provided, in terms

13253 of what key data and data structures represent; what control relationships exist within a subsystem
13254 and how these elements work together to provide the SFR-enforcing behaviour.

13255 The evaluator is reminded that not all tests in the test documentation must map to a subsystem
13256 behaviour or interaction description.

13257 **15.4.1.7.11 Work unit ATE_DPT.1-2**

13258 The evaluator **shall examine** the test plan, test prerequisites, test steps and expected result(s) to
13259 determine that the testing approach for the behaviour description demonstrates the behaviour of
13260 that subsystem as described in the TOE design.

13261 Guidance on this work unit can be found in:

13262 a) 15.2.2, Understanding the expected behaviour of the TOE

13263 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13264 **15.4.1.8 When Evaluation of sub-activity (ATE_DPT.1) is combined with a component of**
13265 **Objectives**

13266 **15.4.1.8** The objectives of this sub-activity are to determine whether the formal security policy model of
13267 the TSF clearly and consistently describes the rules and characteristics of the security policies
13268 and whether this description corresponds with the description of security functions in the
13269 functional specification.

13270 **15.4.1.8 Input**

13271 **15.4.1.8** The evaluation evidence for this sub-activity is:

13272 **15.4.1.8** the ST;

13273 **15.4.1.8** the functional specification;

13274 **15.4.1.8** formal security policy model (ADV_SPM.1.1D);

13275 **15.4.1.8** formal proof of correspondence between the model and any formal functional specification
13276 (ADV_SPM.1.3D);

13277 **15.4.1.8** demonstration of correspondence between the model and the functional specification
13278 (ADV_SPM.1.4D).

13279 **15.4.1.8 Application notes**

13280 **15.4.1.8** This activity applies to cases where the developer has provided a formal security policy model of
13281 the TOE.

13282 **15.4.1.8** A formal TOE security policy model is a representation of the rules (synonymously termed
13283 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
13284 Their formal counterparts are called security properties and security features, respectively. The
13285 representation includes but is not limited to algebraic specifications, finite state machines and
13286 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
13287 model is accompanied by an informal interpretation explaining how the rules and characteristics
13288 are mapped to the respective properties and features.

13289 **15.4.1.8** The creation of a formal security policy model helps to identify and eliminate ambiguous,
13290 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
13291 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement

- 13292 of how well the developer has understood the security functionality being implemented and
 13293 whether there are inconsistencies between the security requirements and the TOE design. The
 13294 confidence in the model is accompanied by a proof that it contains no inconsistencies.
- 13295 **15.4.1.8** A formal security model is a precise formal presentation of the important aspects of
 13296 security and their relationship to the behaviour of the TOE; it identifies the set of rules
 13297 (principles) that defines the TOE security policy and the set of practises (characteristics) that
 13298 regulates how the TSF manages, protects, and otherwise controls the system resources. The
 13299 model includes the set of restrictions and properties that specify how information and computing
 13300 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
 13301 engineering arguments showing that these restrictions and properties play a key role in the
 13302 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
 13303 as well as ancillary text to explain the model and to provide it with context. The security
 13304 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
 13305 with the rest of the TOE and with its operational environment), as well as its internal behaviour.
- 13306 **15.4.1.8** The Security Policy Model of the TOE is informally abstracted from its realisation by
 13307 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 13308 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 13309 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 13310 are always prone to fallacies; especially if relationships among subjects, objects and operations
 13311 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 13312 characteristics of the security policy model are mapped to respective properties and features
 13313 within some formal system, whose rigour and strength can afterwards be used to obtain the
 13314 security properties by means of theorems and formal proof.
- 13315 **15.4.1.8** While the term "formal security policy model" is used in academic circles, the CC's
 13316 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
 13317 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 13318 of SFRs being claimed.
- 13319 **15.4.1.8** The term security policy has traditionally been associated with only access control
 13320 policies, whether label-based (mandatory access control) or user-based (discretionary access
 13321 control). However, a security policy is not limited to access control; there are also audit policies,
 13322 identification policies, authentication policies, encryption policies, management policies, and any
 13323 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 13324 contains an assignment for identifying these policies that are formally modelled.
- 13325 **15.4.1.8** It is recognized that not all policies can be formally modelled for all TOEs. This is
 13326 because either a given policy can not be formally modelled in the otherwise well suited
 13327 framework, or because the nature of the TOE renders impossible the modelling of policies that
 13328 would otherwise be possible to model.
- 13329 **15.4.1.8 Action ADV_SPM.1.1E**
- 13330 **15.4.1.8 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 13331 *text as required, and identify the security policies of the TSF that are modelled.*
- 13332 **15.4.1.8 Work unit ADV_SPM.1-1**
- 13333 **15.4.1.8** The evaluator *shall examine the TOE security policy model to determine that it is*
 13334 *written in a formal style.*
- 13335 **15.4.1.8** The evaluator identifies the formal framework upon which the TOE security policy
 13336 model is based and ensures that it is founded on well established mathematical concepts. **They**
 13337 **also identify the security properties and features addressed in the application notes and ensure**
 13338 **the formalization of at least one security policy.**

- 13339 **15.4.1.8** For guidance on formal methods refer to ISO/IEC **15408-3**
- 13340 **15.4.1.8 Work unit ADV_SPM.1-2**
- 13341 **15.4.1.8** The evaluator *shall examine the TOE security policy model to determine that it*
 13342 *contains all necessary informal explanatory text.*
- 13343 **15.4.1.8** Supporting narrative descriptions are necessary for all parts of the model (for example,
 13344 to make clear the meaning of any formal notation and how they are used) including the security
 13345 properties and features.
- 13346 **15.4.1.8 Work unit ADV_SPM.1-3**
- 13347 **15.4.1.8** The evaluator *shall examine the TOE security policy model to determine that all*
 13348 *security policies of the TSF are identified that are modelled.*
- 13349 **15.4.1.8** The evaluator determines whether the SPM identifies the security policies for which a
 13350 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 13351 of the modelled policies.
- 13352 **15.4.1.8** The evaluator determines whether the list of security policies identified by the SPM is
 13353 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 13354 **15.4.1.8** The evaluator determines whether for each security policy identified by the SPM a
 13355 model is in fact provided.
- 13356 **15.4.1.8 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 13357 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 13358 *not secure.*
- 13359 **15.4.1.8 Work unit ADV_SPM.1-4**
- 13360 **15.4.1.8** The evaluator *shall examine the principles and characteristics of the security policies*
 13361 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 13362 **15.4.1.8** The security policies are expressed in terms of security principles (rules) which are
 13363 modelled by security properties and define the secure state of the TOE. For example, a model
 13364 based on state transitions could describe the security policies in terms of principles of its states,
 13365 identify its initial state, and define what it means to be a secure state.
- 13366 **15.4.1.8** The evaluator determines that the security policies are reflected within their formal
 13367 counterparts of the TSP model.
- 13368 **15.4.1.8** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 13369 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 13370 resources including attributes and conditions of the TOE) which are modelled by security
 13371 features. For example, a model based on state transitions could describe the characteristics as
 13372 possible actions in each secure state in a level of detail sufficient to decide into which state the
 13373 TOE will be transformed by that action.
- 13374 **15.4.1.8** Together the security principles and characteristics describe the entire security posture
 13375 of the TOE.
- 13376 **15.4.1.8** In the context of a formal TOE security policy model the security behaviour is
 13377 considered to be clearly articulated only if an adequate mapping from principles and
 13378 characteristics to their respective formal counterparts properties and features has been given.
 13379 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is

13380	detailed enough to allow for correct identification of all security objectives and the relation to the
13381	security environment.
13382	15.4.1.8 The above condition for clear articulation is necessary but not sufficient. An informal
13383	interpretation of all formal concepts (including attributes, predicates and variables, if available)
13384	must be provided in order to make clear their intended meaning.
13385	15.4.1.8 Work unit ADV_SPM.1-5
13386	15.4.1.8 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
13387	it formally proves that the security features enforce the security properties.
13388	15.4.1.8 To determine the enforcement, the evaluator considers the security properties and the
13389	security features and verifies that the arguments used in the proof are valid. The proof of
13390	correspondence between the security properties and the security features shall be formal.
13391	15.4.1.8 The validity of the security properties shall mean that the TOE is in a secure state. By
13392	this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
13393	state.
13394	15.4.1.8 Work unit ADV_SPM.1-6
13395	15.4.1.8 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
13396	it proves the internal consistency of the TOE security policy model.
13397	15.4.1.8 The proof shall show the absence of contradictions within the TOE security policy
13398	model. In determining the absence of contradictions, the evaluator verifies that the arguments
13399	used in the proof are valid.
13400	15.4.1.8 Since the TOE security policy model is formal, the proof of its internal consistency shall
13401	be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
13402	security policy model usually is not possible due to the fundamental nature of formal frameworks.
13403	Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
13404	security policy model that prove the internal consistency by means of a combination with generic
13405	arguments of the formal framework.
13406	15.4.1.8 ADV_SPM.1.3C <i>The correspondence between the model and the functional</i>
13407	<i>specification shall be at the correct level of formality.</i>
13408	15.4.1.8 Work unit ADV_SPM.1-7
13409	15.4.1.8 The evaluator <i>shall examine the correspondence between the model and the functional</i>
13410	specification to determine that a semiformal demonstration of correspondence between the
13411	model and any semiformal functional specification is provided.
13412	15.4.1.8 This work unit is only applicable to a semiformal presentation of the functional
13413	specification, which is required by ADV_FSP.5.2C.
13414	15.4.1.8 A semiformal correspondence is one that results from a structured approach with a
13415	substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
13416	mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
13417	terms, and so it provides less ambiguity than would exist in an informal correspondence.
13418	15.4.1.8 For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
13419	methods’.

- 13420 **15.4.1.8 Work unit ADV_SPM.1-8**
- 13421 **15.4.1.8** The evaluator *shall examine the correspondence between the model and the*
 13422 functional specification to determine that a formal proof of correspondence between the model
 13423 and any formal functional specification is provided.
- 13424 **15.4.1.8** This work unit is only applicable to a formal presentation of the functional specification,
 13425 which is required by ADV_FSP.6.2D.
- 13426 **15.4.1.8** There should be a formal proof of correspondence between the model and any formal
 13427 functional specification.
- 13428 **15.4.1.8** The formal proof of correspondence removes all subjective interpretations of its terms
 13429 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 13430 formal notation and uses rules that support logical reasoning. The security features within the
 13431 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 13432 language and shown to be satisfied by the formal specification.
- 13433 **15.4.1.8** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 13434 **15.4.1.8 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 13435 *specification is consistent and complete with respect to the model.*
- 13436 **15.4.1.8 Work unit ADV_SPM.1-9**
- 13437 **15.4.1.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 13438 TSF interfaces (as articulated in the functional specification) is complete with respect to the
 13439 behaviour modelled by the security features.
- 13440 **15.4.1.8** The term “correspondence” here means both the formal proof of correspondence
 13441 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 13442 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 13443 **15.4.1.8** In determining completeness of the correspondence, the evaluator considers the
 13444 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 13445 features of the TSP model. The demonstration should show that all characteristics belonging to
 13446 policies that are required to be modelled have an associated feature description in the TOE
 13447 security policy model, and that each feature of the TSP model does occur in the mapping.
- 13448 **15.4.1.8** Abstention from formally modelling TSFI behaviour always calls for justification on the
 13449 developer’s side (also confer the application notes above).
- 13450 **15.4.1.8 Work unit ADV_SPM.1-10**
- 13451 **15.4.1.8** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 13452 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 13453 behaviour modelled by the security features.
- 13454 **15.4.1.8** The term “correspondence” here means both the formal proof of correspondence
 13455 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 13456 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 13457 **15.4.1.8** The meaning of consistency reflects the conventional understanding in contrast to the
 13458 internal consistency concept of work unit ADV_SPM.1-6.
- 13459 **15.4.1.8** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 13460 security features established in the preceding work unit and verifies that the correspondence

- 13461 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
13462 behaviour.
- 13463 **15.4.1.8** For example, if TSFI behaviour dealt with access management on the granularity of
13464 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
13465 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
13466 management for groups of users, then a TSP model describing the security behaviour of the TOE
13467 in terms of individual users would also not be consistent.
- 13468 **15.4.1.8** As another example, if remote untrusted users had to pass more stringent
13469 authentication procedures than administrators whose only point of access were within a
13470 physically-protected area, then this difference in authentication procedures had to be reflected in
13471 the security features.
- 13472 **15.4.1.8** TOE design (ADV_TDS), which includes descriptions at the module level (e.g. Evaluation
13473 of sub-activity (ADV_TDS.3)), the level of detail needed to map the test cases to the behaviour of the
13474 subsystems may require information from the module description to be used. This is because
13475 Evaluation of sub-activity (ADV_TDS.3) allows the description of details to be shifted from the
13476 subsystem level to the module level, or even to omit the subsystems altogether.
- 13477 In any case, the required level of detail in the provided reference to the tested behaviour can be
13478 defined as “the level of detail required for the description of subsystem behaviour as defined by
13479 Evaluation of sub-activity (ADV_TDS.2) (in particular work unit ADV_TDS.2-4)”. It states that a
13480 detailed description of the behaviour typically discusses how the functionality is provided, in terms
13481 of what key data and data structures represent; what control relationships exist within a subsystem
13482 and how these elements work together to provide the SFR-enforcing behaviour.
- 13483 If TSF subsystem interfaces are described, the behaviour of those subsystems may be tested
13484 directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the
13485 TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the
13486 evaluator will consider its appropriateness for adequately testing the behaviour that is described
13487 in the TOE design.
- 13488 **15.4.1.11.11 Work unit ATE_DPT.1-3**
- 13489 The evaluator **shall examine** the test plan, test prerequisites, test steps and expected result(s) to
13490 determine that the testing approach for the behaviour description demonstrates the interactions
13491 among subsystems as described in the TOE design.
- 13492 While the previous work unit addresses behaviour of subsystems, this work unit addresses the
13493 interactions among subsystems.
- 13494 Guidance on this work unit can be found in:
- 13495 a) 15.2.2, Understanding the expected behaviour of the TOE
- 13496 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality
- 13497 If TSF subsystem interfaces are described, the interactions with other subsystems may be tested
13498 directly from those interfaces. Otherwise, the interactions among subsystems must be inferred
13499 from the TSFI interfaces. Whatever strategy is used the evaluator will consider its appropriateness
13500 for adequately testing the interactions among subsystems that are described in the TOE design.
- 13501 ISO/IEC 15408-3 ATE_DPT.1.2C: *The analysis of the depth of testing shall demonstrate that all TSF*
13502 *subsystems in the TOE design have been tested.*

13503 **15.4.1.11.12 Work unit ATE_DPT.1-4**

13504 The evaluator ***shall examine*** the test procedures to determine that all descriptions of TSF
 13505 subsystem behaviour and interaction are tested.

13506 This work unit verifies the completeness of work unit ATE_DPT.1-1. All descriptions of TSF
 13507 subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE
 13508 design have to be tested. Incomplete depth of testing would be evident if a description of TSF
 13509 subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design
 13510 and no tests could be attributed to it.

13511 **15.4.1.12 When Evaluation of sub-activity (ATE_DPT.1) is combined with a component of**
 13512 **Objectives**

13513 **15.4.1.12** The objectives of this sub-activity are to determine whether the formal security policy model of
 13514 the TSF clearly and consistently describes the rules and characteristics of the security policies
 13515 and whether this description corresponds with the description of security functions in the
 13516 functional specification.

13517 **15.4.1.12 Input**

13518 **15.4.1.12** The evaluation evidence for this sub-activity is:

13519 **15.4.1.12** the ST;

13520 **15.4.1.12** the functional specification;

13521 **15.4.1.12** formal security policy model (ADV_SPM.1.1D);

13522 **15.4.1.12** formal proof of correspondence between the model and any formal functional specification
 13523 (ADV_SPM.1.3D);

13524 **15.4.1.12** demonstration of correspondence between the model and the functional specification
 13525 (ADV_SPM.1.4D).

13526 **15.4.1.12 Application notes**

13527 **15.4.1.12** This activity applies to cases where the developer has provided a formal security policy model of
 13528 the TOE.

13529 **15.4.1.12** A formal TOE security policy model is a representation of the rules (synonymously termed
 13530 “principles”) of security policies and characteristics of the TSF behaviour in mathematical terms.
 13531 Their formal counterparts are called security properties and security features, respectively. The
 13532 representation includes but is not limited to algebraic specifications, finite state machines and
 13533 logic formalisms strong enough to formally infer the properties from the features. The formal TSP
 13534 model is accompanied by an informal interpretation explaining how the rules and characteristics
 13535 are mapped to the respective properties and features.

13536 **15.4.1.12** The creation of a formal security policy model helps to identify and eliminate ambiguous,
 13537 inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
 13538 built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
 13539 of how well the developer has understood the security functionality being implemented and
 13540 whether there are inconsistencies between the security requirements and the TOE design. The
 13541 confidence in the model is accompanied by a proof that it contains no inconsistencies.

13542 **15.4.1.12** A formal security model is a precise formal presentation of the important aspects of security and
 13543 their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that
 13544 defines the TOE security policy and the set of practises (characteristics) that regulates how the

- 13545 TSF manages, protects, and otherwise controls the system resources. The model includes the set
13546 of restrictions and properties that specify how information and computing resources are
13547 prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
13548 arguments showing that these restrictions and properties play a key role in the enforcement of
13549 the SFRs. It consists both of the formalisms that express the security functionality, as well as
13550 ancillary text to explain the model and to provide it with context. The security behaviour of the
13551 TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
13552 the TOE and with its operational environment), as well as its internal behaviour.
- 13553 **15.4.1.12** The Security Policy Model of the TOE is informally abstracted from its realisation by
13554 considering the proposed security requirements of the ST. The informal abstraction is taken to be
13555 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
13556 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
13557 are always prone to fallacies; especially if relationships among subjects, objects and operations
13558 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
13559 characteristics of the security policy model are mapped to respective properties and features
13560 within some formal system, whose rigour and strength can afterwards be used to obtain the
13561 security properties by means of theorems and formal proof.
- 13562 **15.4.1.12** While the term “formal security policy model” is used in academic circles, the CC's
13563 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
13564 claimed. Therefore, the formal security policy model is merely a formal representation of the set
13565 of SFRs being claimed.
- 13566 **15.4.1.12** The term security policy has traditionally been associated with only access control
13567 policies, whether label-based (mandatory access control) or user-based (discretionary access
13568 control). However, a security policy is not limited to access control; there are also audit policies,
13569 identification policies, authentication policies, encryption policies, management policies, and any
13570 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
13571 contains an assignment for identifying these policies that are formally modelled.
- 13572 **15.4.1.12** It is recognized that not all policies can be formally modelled for all TOEs. This is
13573 because either a given policy can not be formally modelled in the otherwise well suited
13574 framework, or because the nature of the TOE renders impossible the modelling of policies that
13575 would otherwise be possible to model.
- 13576 **15.4.1.12 Action ADV_SPM.1.1E**
- 13577 **15.4.1.12 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
13578 *text as required, and identify the security policies of the TSF that are modelled.*
- 13579 **15.4.1.12 Work unit ADV_SPM.1-1**
- 13580 **15.4.1.12** The evaluator ***shall examine the TOE security policy model to determine that it is***
13581 **written in a formal style.**
- 13582 **15.4.1.12** The evaluator identifies the formal framework upon which the TOE security policy
13583 model is based and ensures that it is founded on well established mathematical concepts. **They**
13584 **also identify the security properties and features addressed in the application notes and ensure**
13585 **the formalization of at least one security policy.**
- 13586 **15.4.1.12** For guidance on formal methods refer to ISO/IEC **15408-3**
- 13587 **15.4.1.12 Work unit ADV_SPM.1-2**
- 13588 **15.4.1.12** The evaluator ***shall examine the TOE security policy model to determine that it***
13589 **contains all necessary informal explanatory text.**

- 13590 **15.4.1.12** Supporting narrative descriptions are necessary for all parts of the model (for
13591 example, to make clear the meaning of any formal notation and how they are used) including the
13592 security properties and features.
- 13593 **15.4.1.12 Work unit ADV_SPM.1-3**
- 13594 **15.4.1.12** The evaluator *shall examine the TOE security policy model to determine that all*
13595 security policies of the TSF are identified that are modelled.
- 13596 **15.4.1.12** The evaluator determines whether the SPM identifies the security policies for which a
13597 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
13598 of the modelled policies.
- 13599 **15.4.1.12** The evaluator determines whether the list of security policies identified by the SPM is
13600 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 13601 **15.4.1.12** The evaluator determines whether for each security policy identified by the SPM a
13602 model is in fact provided.
- 13603 **15.4.1.12 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
13604 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
13605 *not secure.*
- 13606 **15.4.1.12 Work unit ADV_SPM.1-4**
- 13607 **15.4.1.12** The evaluator *shall examine the principles and characteristics of the security policies*
13608 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 13609 **15.4.1.12** The security policies are expressed in terms of security principles (rules) which are
13610 modelled by security properties and define the secure state of the TOE. For example, a model
13611 based on state transitions could describe the security policies in terms of principles of its states,
13612 identify its initial state, and define what it means to be a secure state.
- 13613 **15.4.1.12** The evaluator determines that the security policies are reflected within their formal
13614 counterparts of the TSP model.
- 13615 **15.4.1.12** The TOE security behaviour is expressed in terms of security characteristics (i.e.
13616 portions of TOE security functionality managing, protecting, and otherwise controlling the system
13617 resources including attributes and conditions of the TOE) which are modelled by security
13618 features. For example, a model based on state transitions could describe the characteristics as
13619 possible actions in each secure state in a level of detail sufficient to decide into which state the
13620 TOE will be transformed by that action.
- 13621 **15.4.1.12** Together the security principles and characteristics describe the entire security
13622 posture of the TOE.
- 13623 **15.4.1.12** In the context of a formal TOE security policy model the security behaviour is
13624 considered to be clearly articulated only if an adequate mapping from principles and
13625 characteristics to their respective formal counterparts properties and features has been given.
13626 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
13627 detailed enough to allow for correct identification of all security objectives and the relation to the
13628 security environment.
- 13629 **15.4.1.12** The above condition for clear articulation is necessary but not sufficient. An informal
13630 interpretation of all formal concepts (including attributes, predicates and variables, if available)
13631 must be provided in order to make clear their intended meaning.

- 13632 **15.4.1.12 Work unit ADV_SPM.1-5**
- 13633 **15.4.1.12** The evaluator *shall examine the TOE security policy model rationale to*
 13634 determine that it formally proves that the security features enforce the security properties.
- 13635 **15.4.1.12** To determine the enforcement, the evaluator considers the security properties and
 13636 the security features and verifies that the arguments used in the proof are valid. The proof of
 13637 correspondence between the security properties and the security features shall be formal.
- 13638 **15.4.1.12** The validity of the security properties shall mean that the TOE is in a secure state. By
 13639 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 13640 state.
- 13641 **15.4.1.12 Work unit ADV_SPM.1-6**
- 13642 **15.4.1.12** The evaluator *shall examine the TOE security policy model rationale to determine*
 13643 that it proves the internal consistency of the TOE security policy model.
- 13644 **15.4.1.12** The proof shall show the absence of contradictions within the TOE security policy
 13645 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 13646 used in the proof are valid.
- 13647 **15.4.1.12** Since the TOE security policy model is formal, the proof of its internal consistency
 13648 shall be formal. It is recognized that a complete formal proof of the internal consistency of the
 13649 TOE security policy model usually is not possible due to the fundamental nature of formal
 13650 frameworks. Generally, it is sufficient to generate evidence using formal proofs based on the
 13651 specific TOE security policy model that prove the internal consistency by means of a combination
 13652 with generic arguments of the formal framework.
- 13653 **15.4.1.12 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 13654 *specification shall be at the correct level of formality.*
- 13655 **15.4.1.12 Work unit ADV_SPM.1-7**
- 13656 **15.4.1.12** The evaluator *shall examine the correspondence between the model and the*
 13657 *functional specification to determine that a semiformal demonstration of correspondence*
 13658 *between the model and any semiformal functional specification is provided.*
- 13659 **15.4.1.12** This work unit is only applicable to a semiformal presentation of the functional
 13660 specification, which is required by ADV_FSP.5.2C.
- 13661 **15.4.1.12** A semiformal correspondence is one that results from a structured approach with a
 13662 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 13663 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 13664 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 13665 **15.4.1.12** For guidance on semiformal methods refer to Annex 3.1.1 ‘**Semiformal and formal**
 13666 **methods**’.
- 13667 **15.4.1.12 Work unit ADV_SPM.1-8**
- 13668 **15.4.1.12** The evaluator *shall examine the correspondence between the model and the*
 13669 *functional specification to determine that a formal proof of correspondence between the model*
 13670 *and any formal functional specification is provided.*
- 13671 **15.4.1.12** This work unit is only applicable to a formal presentation of the functional
 13672 specification, which is required by ADV_FSP.6.2D.

- 13673 **15.4.1.12** There should be a formal proof of correspondence between the model and any formal
13674 functional specification.
- 13675 **15.4.1.12** The formal proof of correspondence removes all subjective interpretations of its
13676 terms by enlisting well-established mathematical concepts to define the syntax and semantics of
13677 the formal notation and uses rules that support logical reasoning. The security features within
13678 the TOE (which are identified in the formal TSP model) are expressed in a formal specification
13679 language and shown to be satisfied by the formal specification.
- 13680 **15.4.1.12** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 13681 **15.4.1.12 ADV_SPM.1.4C** *The correspondence shall show that the functional*
13682 *specification is consistent and complete with respect to the model.*
- 13683 **15.4.1.12 Work unit ADV_SPM.1-9**
- 13684 **15.4.1.12** The evaluator ***shall examine the correspondence to determine that the behaviour at***
13685 ***the TSF interfaces (as articulated in the functional specification) is complete with respect to the***
13686 ***behaviour modelled by the security features.***
- 13687 **15.4.1.12** The term “correspondence” here means both the formal proof of correspondence
13688 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
13689 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 13690 **15.4.1.12** In determining completeness of the correspondence, the evaluator considers the
13691 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
13692 features of the TSP model. The demonstration should show that all characteristics belonging to
13693 policies that are required to be modelled have an associated feature description in the TOE
13694 security policy model, and that each feature of the TSP model does occur in the mapping.
- 13695 **15.4.1.12** Abstention from formally modelling TSFI behaviour always calls for justification on
13696 the developer’s side (also confer the application notes above).
- 13697 **15.4.1.12 Work unit ADV_SPM.1-10**
- 13698 **15.4.1.12** The evaluator ***shall examine the correspondence to determine that the behaviour at***
13699 ***the TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
13700 ***behaviour modelled by the security features.***
- 13701 **15.4.1.12** The term “correspondence” here means both the formal proof of correspondence
13702 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
13703 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 13704 **15.4.1.12** The meaning of consistency reflects the conventional understanding in contrast to the
13705 internal consistency concept of work unit ADV_SPM.1-6.
- 13706 **15.4.1.12** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
13707 security features established in the preceding work unit and verifies that the correspondence
13708 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
13709 behaviour.
- 13710 **15.4.1.12** For example, if TSFI behaviour dealt with access management on the granularity of
13711 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
13712 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
13713 management for groups of users, then a TSP model describing the security behaviour of the TOE
13714 in terms of individual users would also not be consistent.

13715 **15.4.1.12** As another example, if remote untrusted users had to pass more stringent
 13716 authentication procedures than administrators whose only point of access were within a
 13717 physically-protected area, then this difference in authentication procedures had to be reflected in
 13718 the security features.

13719 **15.4.1.12** TOE design (ADV_TDS), which includes descriptions at the module level (e.g.
 13720 Evaluation of sub-activity (ADV_TDS.3)), the level of detail needed to map the test cases to the
 13721 behaviour of the subsystems may require information from the module description to be used. This
 13722 is because Evaluation of sub-activity (ADV_TDS.3) allows the description of details to be shifted
 13723 from the subsystem level to the module level, or even to omit the subsystems altogether.

13724 In any case, the required level of detail in the provided reference to the tested behaviour can be
 13725 defined as “the level of detail required for the description of subsystem behaviour as defined by
 13726 Evaluation of sub-activity (ADV_TDS.2) (in particular work unit ADV_TDS.2-4)”. It states that a
 13727 detailed description of the behaviour typically discusses how the functionality is provided, in terms
 13728 of what key data and data structures represent; what control relationships exist within a subsystem
 13729 and how these elements work together to provide the SFR-enforcing behaviour.

13730 The evaluator is reminded that this does not imply that all tests in the test documentation must
 13731 map to the subsystem behaviour or interaction description in the TOE design.

13732 **15.4.2 Evaluation of sub-activity (ATE_DPT.2)**

13733 **15.4.2.1 Objectives**

13734 The objective of this sub-activity is to determine whether the developer has tested all the TSF
 13735 subsystems and SFR-enforcing modules against the TOE design and the security architecture
 13736 description.

13737 **15.4.2.2 Input**

- 13738 a) the ST;
- 13739 b) the functional specification;
- 13740 c) the TOE design;
- 13741 d) the security architecture description;
- 13742 e) the test documentation;
- 13743 f) the depth of testing analysis.

13744 **15.4.2.3 Action ATE_DPT.2.1E**

13745 ISO/IEC 15408-3 ATE_DPT.2.1C: *The analysis of the depth of testing shall demonstrate the*
 13746 *correspondence between the tests in the test documentation and the TSF subsystems and SFR-*
 13747 *enforcing modules in the TOE design.*

13748 **15.4.2.3.1 Work unit ATE_DPT.2-1**

13749 The evaluator ***shall examine*** the depth of testing analysis to determine that descriptions of the
 13750 behaviour of TSF subsystems and of their interactions are included within the test documentation.

13751 This work unit verifies the content of the correspondence between the tests and the descriptions in
 13752 the TOE design. In cases where the description of the TSF's architectural soundness (in Security
 13753 Architecture (ADV_ARC)) cites specific mechanisms, this work unit also verifies the
 13754 correspondence between the tests and the descriptions of the behaviour of such mechanisms.

13755 A simple cross-table may be sufficient to show test correspondence. The identification of the tests
13756 and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

13757 The evaluator is reminded that not all tests in the test documentation must map to a subsystem
13758 behaviour or interaction description.

13759 **15.4.2.3.2 Work unit ATE_DPT.2-2**

13760 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to
13761 determine that the testing approach for the behaviour description demonstrates the behaviour of
13762 that subsystem as described in the TOE design.

13763 Guidance on this work unit can be found in:

13764 a) 15.2.2, Understanding the expected behaviour of the TOE

13765 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13766 If TSF subsystem interfaces are described, the behaviour of those subsystems may be tested
13767 directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the
13768 TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the
13769 evaluator will consider its appropriateness for adequately testing the behaviour that is described
13770 in the TOE design.

13771 **15.4.2.3.3 Work unit ATE_DPT.2-3**

13772 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to
13773 determine that the testing approach for the behaviour description demonstrates the interactions
13774 among subsystems as described in the TOE design.

13775 While the previous work unit addresses behaviour of subsystems, this work unit addresses the
13776 interactions among subsystems.

13777 Guidance on this work unit can be found in:

13778 a) 15.2.2, Understanding the expected behaviour of the TOE

13779 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13780 If TSF subsystem interfaces are described, the interactions with other subsystems may be tested
13781 directly from those interfaces. Otherwise, the interactions among subsystems must be inferred
13782 from the TSFI interfaces. Whatever strategy is used the evaluator will consider its appropriateness
13783 for adequately testing the interactions among subsystems that are described in the TOE design.

13784 **15.4.2.3.4 Work unit ATE_DPT.2-4**

13785 The evaluator ***shall examine*** the depth of testing analysis to determine that the interfaces of SFR-
13786 enforcing modules are included within the test documentation.

13787 This work unit verifies the content of the correspondence between the tests and the descriptions in
13788 the TOE design. In cases where the description of the TSF's architectural soundness (in Security
13789 Architecture (ADV_ARC)) cites specific mechanisms at the modular level, this work unit also
13790 verifies the correspondence between the tests and the descriptions of the behaviour of such
13791 mechanisms.

13792 A simple cross-table may be sufficient to show test correspondence. The identification of the tests
13793 and the SFR-enforcing modules presented in the depth-of coverage analysis has to be unambiguous.

13794 The evaluator is reminded that not all tests in the test documentation must map to the interfaces of
13795 SFR-enforcing modules.

13796 **15.4.2.3.5 Work unit ATE_DPT.2-5**

13797 The evaluator *shall examine* the test plan, test prerequisites, test steps and expected result(s) to
13798 determine that the testing approach for each SFR-enforcing module interface demonstrates the
13799 expected behaviour of that interface.

13800 While work unit ATE_DPT.2-2 addresses expected behaviour of subsystems, this work unit
13801 addresses expected behaviour of the SFR-enforcing module interfaces that are covered by
13802 ATE_DPT.2-4.

13803 Guidance on this work unit can be found in:

13804 a) 15.2.2, Understanding the expected behaviour of the TOE

13805 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13806 Testing of an interface may be performed directly at that interface, or at the external interfaces, or
13807 a combination of both. Whatever strategy is used the evaluator will consider its appropriateness
13808 for adequately testing the interfaces. Specifically the evaluator determines whether testing at the
13809 internal interfaces is necessary or whether these internal interfaces can be adequately tested
13810 (albeit implicitly) by exercising the external interfaces. This determination is left to the evaluator,
13811 as is its justification.

13812 ISO/IEC 15408-3 ATE_DPT.2.2C: *The analysis of the depth of testing shall demonstrate that all TSF*
13813 *subsystems in the TOE design have been tested.*

13814 **15.4.2.3.6 Work unit ATE_DPT.2-6**

13815 The evaluator *shall examine* the test procedures to determine that all descriptions of TSF
13816 subsystem behaviour and interaction are tested.

13817 This work unit verifies the completeness of work unit ATE_DPT.2-1. All descriptions of TSF
13818 subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE
13819 design have to be tested. Incomplete depth of testing would be evident if a description of TSF
13820 subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design
13821 and no tests could be attributed to it.

13822 The evaluator is reminded that this does not imply that all tests in the test documentation must
13823 map to the subsystem behaviour or interaction description in the TOE design.

13824 ISO/IEC 15408-3 ATE_DPT.2.3C: *The analysis of the depth of testing shall demonstrate that the SFR-*
13825 *enforcing modules in the TOE design have been tested.*

13826 **15.4.2.3.7 Work unit ATE_DPT.2-7**

13827 The evaluator *shall examine* the test procedures to determine that all interfaces of SFR-enforcing
13828 modules are tested.

13829 This work unit verifies the completeness of work unit ATE_DPT.2-4. All interfaces of SFR-enforcing
13830 modules that are provided in the TOE design have to be tested. Incomplete depth of testing would
13831 be evident if any interface of any SFR-enforcing modules was identified in the TOE design and no
13832 tests could be attributed to it.

13833 The evaluator is reminded that this does not imply that all tests in the test documentation must
13834 map to an interface of an SFR-enforcing module in the TOE design.

13835 **15.4.3 Evaluation of sub-activity (ATE_DPT.3)**

13836 **15.4.3.1 Objectives**

13837 The objective of this sub-activity is to determine whether the developer has tested the all the TSF
13838 subsystems and modules against the TOE design and the security architecture description.

13839 **15.4.3.2 Input**

- 13840 a) the ST;
- 13841 b) the functional specification;
- 13842 c) the TOE design;
- 13843 d) the security architecture description;
- 13844 e) the test documentation;
- 13845 f) the depth of testing analysis.

13846 **15.4.3.3 Action ATE_DPT.3.1E**

13847 ISO/IEC 15408-3 ATE_DPT.3.1C: *The analysis of the depth of testing shall demonstrate the*
13848 *correspondence between the tests in the test documentation and the TSF subsystems and modules in*
13849 *the TOE design.*

13850 **15.4.3.3.1 Work unit ATE_DPT.3-1**

13851 The evaluator ***shall examine*** the depth of testing analysis to determine that descriptions of the
13852 behaviour of TSF subsystems and of their interactions are included within the test documentation.

13853 This work unit verifies the content of the correspondence between the tests and the descriptions in
13854 the TOE design. A simple cross-table may be sufficient to show test correspondence. The
13855 identification of the tests and the behaviour/interaction presented in the depth-of coverage
13856 analysis has to be unambiguous.

13857 The evaluator is reminded that not all tests in the test documentation must map to a subsystem
13858 behaviour or interaction description.

13859 **15.4.3.3.2 Work unit ATE_DPT.3-2**

13860 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to
13861 determine that the testing approach for the behaviour description demonstrates the behaviour of
13862 that subsystem as described in the TOE design.

13863 Guidance on this work unit can be found in:

- 13864 a) 15.2.2, Understanding the expected behaviour of the TOE
- 13865 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13866 If TSF subsystem interfaces are provided, the behaviour of those subsystems may be performed
13867 directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the
13868 TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the
13869 evaluator will consider its appropriateness for adequately testing the behaviour that is described
13870 in the TOE design.

13871 **15.4.3.3.3 Work unit ATE_DPT.3-3**

13872 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to
 13873 determine that the testing approach for the behaviour description demonstrates the interactions
 13874 among subsystems as described in the TOE design.

13875 Guidance on this work unit can be found in:

13876 a) 15.2.2, Understanding the expected behaviour of the TOE

13877 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13878 While the previous work unit addresses behaviour of subsystems, this work unit addresses the
 13879 interactions among subsystems.

13880 If TSF subsystem interfaces are provided, the interactions with other subsystems may be
 13881 performed directly from those interfaces. Otherwise, the interactions among subsystems must be
 13882 inferred from the TSFI interfaces. Whatever strategy is used the evaluator will consider its
 13883 appropriateness for adequately testing the interactions among subsystems that are described in
 13884 the TOE design.

13885 **15.4.3.3.4 Work unit ATE_DPT.3-4**

13886 The evaluator ***shall examine*** the depth of testing analysis to determine that the interfaces of TSF
 13887 modules are included within the test documentation.

13888 This work unit verifies the content of the correspondence between the tests and the descriptions in
 13889 the TOE design. A simple cross-table may be sufficient to show test correspondence. The
 13890 identification of the tests and the behaviour/interaction presented in the depth-of coverage
 13891 analysis has to be unambiguous.

13892 The evaluator is reminded that not all tests in the test documentation must map to a subsystem
 13893 behaviour or interaction description.

13894 **15.4.3.3.5 Work unit ATE_DPT.3-5**

13895 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to
 13896 determine that the testing approach for each TSF module interface demonstrates the expected
 13897 behaviour of that interface.

13898 Guidance on this work unit can be found in:

13899 a) 15.2.2, Understanding the expected behaviour of the TOE

13900 b) 15.2.3, Testing vs. alternate approaches to verify the expected behaviour of functionality

13901 Testing of an interface may be performed directly at that interface, or at the external interfaces, or
 13902 a combination of both. Whatever strategy is used the evaluator will consider its appropriateness
 13903 for adequately testing the interfaces. Specifically the evaluator determines whether testing at the
 13904 internal interfaces is necessary or whether these internal interfaces can be adequately tested
 13905 (albeit implicitly) by exercising the external interfaces. This determination is left to the evaluator,
 13906 as is its justification.

13907 ISO/IEC 15408-3 ATE_DPT.3.2C: *The analysis of the depth of testing shall demonstrate that all TSF*
 13908 *subsystems in the TOE design have been tested.*

13909 **15.4.3.3.6 Work unit ATE_DPT.3-6**

13910 The evaluator **shall examine** the test procedures to determine that all descriptions of TSF
13911 subsystem behaviour and interaction are tested.

13912 This work unit verifies the completeness of work unit ATE_DPT.3-1. All descriptions of TSF
13913 subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE
13914 design have to be tested. Incomplete depth of testing would be evident if a description of TSF
13915 subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design
13916 and no tests could be attributed to it.

13917 The evaluator is reminded that this does not imply that all tests in the test documentation must
13918 map to the subsystem behaviour or interaction description in the TOE design.

13919 ISO/IEC 15408-3 ATE_DPT.3.3C: *The analysis of the depth of testing shall demonstrate that all TSF*
13920 *modules in the TOE design have been tested.*

13921 **15.4.3.3.7 Work unit ATE_DPT.3-7**

13922 The evaluator **shall examine** the test procedures to determine that all interfaces of all TSF modules
13923 are tested.

13924 This work unit verifies the completeness of work unit ATE_DPT.3-4. All interfaces of TSF modules
13925 that are provided in the TOE design have to be tested. Incomplete depth of testing would be
13926 evident if any interface of any TSF module was identified in the TOE design and no tests could be
13927 attributed to it.

13928 The evaluator is reminded that this does not imply that all tests in the test documentation must
13929 map to an interface of a TSF module in the TOE design.

13930 **15.4.4 Evaluation of sub-activity (ATE_DPT.4)**

13931 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

13932 **15.5 Functional tests (ATE_FUN)**

13933 **15.5.1 Evaluation of sub-activity (ATE_FUN.1)**

13934 **15.5.1.1 Objectives**

13935 The objective of this sub-activity is to determine whether the developer correctly performed and
13936 documented the tests in the test documentation.

13937 **15.5.1.2 Input**

13938 The evaluation evidence for this sub-activity is:

- 13939 a) the ST;
- 13940 b) the functional specification;
- 13941 c) the test documentation.

13942 **15.5.1.3 Application notes**

13943 The extent to which the test documentation is required to cover the TSF is dependent upon the
13944 coverage assurance component.

13945 For the developer tests provided, the evaluator determines whether the tests are repeatable, and
13946 the extent to which the developer's tests can be used for the evaluator's independent testing effort.
13947 Any TSFI for which the developer's test results indicate that it might not perform as specified
13948 should be tested independently by the evaluator to determine whether or not it does.

13949 **15.5.1.4 Action ATE_FUN.1.1E**

13950 ISO/IEC 15408-3 ATE_FUN.1.1C: *The test documentation shall consist of test plans, expected test*
13951 *results and actual test results.*

13952 **15.5.1.4.1 Work unit ATE_FUN.1-1**

13953 The evaluator ***shall check*** that the test documentation includes test plans, expected test results and
13954 actual test results.

13955 The evaluator checks that test plans, expected tests results and actual test results are included in
13956 the test documentation.

13957 ISO/IEC 15408-3 ATE_FUN.1.2C: *The test plans shall identify the tests to be performed and describe*
13958 *the scenarios for performing each test. These scenarios shall include any ordering dependencies on the*
13959 *results of other tests.*

13960 **15.5.1.4.2 Work unit ATE_FUN.1-2**

13961 The evaluator ***shall examine*** the test plan to determine that it describes the scenarios for
13962 performing each test.

13963 The evaluator determines that the test plan provides information about the test configuration
13964 being used: both on the configuration of the TOE and on any test equipment being used. This
13965 information should be detailed enough to ensure that the test configuration is reproducible.

13966 The evaluator also determines that the test plan provides information about how to execute the
13967 test: any necessary automated set-up procedures (and whether they require privilege to run),
13968 inputs to be applied, how these inputs are applied, how output is obtained, any automated clean-up
13969 procedures (and whether they require privilege to run), etc. This information should be detailed
13970 enough to ensure that the test is reproducible.

13971 The evaluator may wish to employ a sampling strategy when performing this work unit.

13972 **15.5.1.4.3 Work unit ATE_FUN.1-3**

13973 The evaluator ***shall examine*** the test plan to determine that the TOE test configuration is
13974 consistent with the ST.

13975 The TOE referred to in the developer's test plan should have the same unique reference as
13976 established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

13977 It is possible for the ST to specify more than one configuration for evaluation. The evaluator
13978 verifies that all test configurations identified in the developer test documentation are consistent
13979 with the ST. For example, the ST might define configuration options that must be set, which could
13980 have an impact upon what constitutes the TOE by including or excluding additional portions. The
13981 evaluator verifies that all such variations of the TOE are considered.

13982 The evaluator should consider the security objectives for the operational environment described in
13983 the ST that may apply to the test environment. There may be some objectives for the operational
13984 environment that do not apply to the test environment. For example, an objective about user
13985 clearances may not apply; however, an objective about a single point of connection to a network
13986 would apply.

- 13987 The evaluator may wish to employ a sampling strategy when performing this work unit.
- 13988 If this work unit is applied to a component TOE that might be used/integrated in a composed TOE
 13989 (see The work units for the evaluation of the sub-activity AVA_VAN.5 are copied from the work
 13990 units of AVA_VAN.4 as far as possible except that the TOE is attacked by attackers possessing High
 13991 attack potential.
- 13992 **Objectives**
- 13993 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
 13994 has vulnerabilities exploitable by attackers possessing **High** attack potential.
- 13995 **Input**
- 13996 The evaluation evidence for this sub-activity is:
- 13997 the ST;
- 13998 the functional specification;
- 13999 the TOE design;
- 14000 the security architecture description;
- 14001 the implementation representation;
- 14002 the guidance documentation;
- 14003 the TOE suitable for testing;
- 14004 information publicly available to support the identification of possible potential vulnerabilities;
- 14005 the results of the testing of the basic design.
- 14006 The remaining implicit evaluation evidence for this sub-activity depends on the components that
 14007 have been included in the assurance package. The evidence provided for each component is to be
 14008 used as input in this sub-activity.
- 14009 Other input for this sub-activity is:
- 14010 current information regarding public domain potential vulnerabilities and attacks (e.g. from an
 14011 evaluation authority).
- 14012 **Application notes**
- 14013 The methodical analysis approach takes the form of a structured examination of the evidence. This
 14014 method requires the evaluator to specify the structure and form the analysis will take (i.e. the
 14015 manner in which the analysis is performed is predetermined, unlike the focused analysis). The
 14016 method is specified in terms of the information that will be considered and how/why it will be
 14017 considered. Further guidance on methodical vulnerability analysis can be found in Annex B.2.2.2.3.
- 14018 **Action AVA_VAN.5.1E**
- 14019 **AVA_VAN.5.1C**
- 14020 The TOE shall be suitable for testing.

14021 **Work unit AVA_VAN.5-1**

14022 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
14023 the configuration under evaluation as specified in the ST.

14024 The TOE provided by the developer and identified in the test plan should have the same unique
14025 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
14026 introduction.

14027 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
14028 comprise a number of distinct hardware and software entities that need to be tested in accordance
14029 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

14030 The evaluator should consider the security objectives for the operational environment described in
14031 the ST that may apply to the test environment and ensure they are met in the testing environment.
14032 There may be some objectives for the operational environment that do not apply to the test
14033 environment. For example, an objective about user clearances may not apply; however, an
14034 objective about a single point of connection to a network would apply.

14035 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
14036 ensure that these resources are calibrated correctly.

14037 **Work unit AVA_VAN.5-2**

14038 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a
14039 known state

14040 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
14041 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
14042 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
14043 installed properly and is in a known state. If this is not the case, then the evaluator should follow
14044 the developer's procedures to install and start up the TOE, using the supplied guidance only.

14045 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
14046 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.

14047 **Action AVA_VAN.5.2E**

14048 **Work unit AVA_VAN.5-3**

14049 The evaluator ***shall examine*** sources of information publicly available to identify potential
14050 vulnerabilities in the TOE.

14051 The evaluator examines the sources of information publicly available to support the identification
14052 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
14053 information which the evaluator should consider using items such as those available on the world
14054 wide web, including:

14055 specialist publications (magazines, books);

14056 research papers;

14057 conference proceedings.

14058 The evaluator should not constrain their consideration of publicly available information to the
14059 above, but should consider any other relevant information available.

- 14060 While examining the evidence provided the evaluator will use the information in the public domain
 14061 to further search for potential vulnerabilities. Where the evaluators have identified areas of
 14062 concern, the evaluator should consider information publicly available that relate to those areas of
 14063 concern.
- 14064 The availability of information that may be readily available to an attacker that helps to identify
 14065 and facilitate attacks may substantially enhance the attack potential of a given attacker. The
 14066 accessibility of vulnerability information and sophisticated attack tools on the Internet makes it
 14067 more likely that this information will be used in attempts to identify potential vulnerabilities in the
 14068 TOE and exploit them. Modern search tools make such information easily available to the evaluator,
 14069 and the determination of resistance to published potential vulnerabilities and well known generic
 14070 attacks can be achieved in a cost-effective manner.
- 14071 The search of the information publicly available should be focused on those sources that refer to
 14072 the technologies used in the development of the product from which the TOE is derived. The
 14073 extensiveness of this search should consider the following factors: TOE type, evaluator experience
 14074 in this TOE type, expected attack potential and the level of ADV evidence available.
- 14075 The identification process is iterative, where the identification of one potential vulnerability may
 14076 lead to identifying another area of concern that requires further investigation.
- 14077 The evaluator will describe the approach to be taken to identify potential vulnerabilities in the
 14078 publicly available material, detailing the search to be performed. This may be driven by factors
 14079 such as areas of concern identified by the evaluator, linked to the evidence the attacker is assumed
 14080 to be able to obtain. However, it is recognised that in this type of search the approach may further
 14081 evolve as a result of findings during the search. Therefore, the evaluator will also report any
 14082 actions taken in addition to those described in the approach to further investigate issues thought to
 14083 lead to potential vulnerabilities, and will report the evidence examined in completing the search
 14084 for potential vulnerabilities.
- 14085 **Action AVA_VAN.5.3E**
- 14086 **Work unit AVA_VAN.5-4**
- 14087 The evaluator ***shall conduct*** a methodical analysis of ST, guidance documentation, functional
 14088 specification, TOE design, security architecture description and implementation representation to
 14089 identify possible potential vulnerabilities in the TOE.
- 14090 Guidance on methodical vulnerability analysis is provided in Annex B.2.2.2.3.
- 14091 This approach to identification of potential vulnerabilities is to take an ordered and planned
 14092 approach. A system is to be applied in the examination. The evaluator is to describe the method to
 14093 be used in terms of the manner in which this information is to be considered and the hypothesis
 14094 that is to be created.
- 14095 A flaw hypothesis methodology should be used whereby the ST, development (functional
 14096 specification, TOE design and implementation representation) and guidance evidence are analysed
 14097 and then vulnerabilities in the TOE are hypothesised, or speculated.
- 14098 The evaluator should use the knowledge of the TOE design and operation gained from the TOE
 14099 deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE
 14100 and potential errors in the specified method of operation of the TOE.
- 14101 The security architecture description provides the developer vulnerability analysis, as it
 14102 documents how the TSF protects itself from interference from untrusted subjects and prevents the
 14103 bypass of security enforcement functionality. Therefore, the evaluator should build upon the
 14104 understanding of the TSF protection gained from the analysis of this evidence and then develop
 14105 this in the knowledge gained from other development (e.g. ADV) evidence.

- 14106 The approach taken to the methodical search for vulnerabilities is to consider any areas of concern
14107 identified in the results of the evaluator's assessment of the development and guidance evidence.
14108 However, the evaluator should also consider each aspect of the security architecture analysis to
14109 search for any ways in which the protection of the TSF can be undermined. It may be helpful to
14110 structure the methodical analysis on the basis of the material presented in the security architecture
14111 description, introducing concerns from other ADV evidence as appropriate. The analysis can then
14112 be further developed to ensure all other material from the ADV evidence is considered.
- 14113 The following provide some examples of hypotheses that may be created when examining the
14114 evidence:
- 14115 consideration of malformed input for interfaces available to an attacker at the external interfaces;
- 14116 examination of a key security mechanism cited in the security architecture description, such as
14117 process separation, hypothesising internal buffer overflows that may lead to degradation of
14118 separation;
- 14119 search to identify any objects created in the TOE implementation representation that are then not
14120 fully controlled by the TSF, and could be used by an attacker to undermine SFRs.
- 14121 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE
14122 and specify an approach to the search that 'all interface specifications in the evidence provided will
14123 be searched to hypothesise potential vulnerabilities' and go on to explain the methods used in the
14124 hypothesis.
- 14125 In addition, areas of concern the evaluator has identified during examination of the evidence
14126 during the conduct of evaluation activities. Areas of concern may also be identified during the
14127 conduct of other work units associated with this component, in particular AVA_VAN.5-7,
14128 AVA_VAN.5-5 and AVA_VAN.5-6) where the development and conduct of penetration tests may
14129 identify further areas of concerns for investigation, or potential vulnerabilities.
- 14130 However, examination of only a subset of the development and guidance evidence or their contents
14131 is not permitted in this level of rigour. The approach description should provide a demonstration
14132 that the methodical approach used is complete, providing confidence that the approach used to
14133 search the deliverables has considered all of the information provided in those deliverables.
- 14134 This approach to identification of potential vulnerabilities is to take an ordered and planned
14135 approach; applying a system to the examination. The evaluator is to describe the method to be used
14136 in terms of how the evidence will be considered; the manner in which this information is to be
14137 considered and the hypothesis that is to be created. This approach should be agreed with the
14138 evaluation authority, and the evaluation authority should provide detail of any additional
14139 approaches the evaluator should take to the vulnerability analysis and identify any additional
14140 information that should be considered by the evaluator.
- 14141 Although a system to identifying potential vulnerabilities is predefined, the identification process
14142 may still be iterative, where the identification of one potential vulnerability may lead to identifying
14143 another area of concern that requires further investigation.
- 14144 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent
14145 vulnerability analysis should consider generic potential vulnerabilities under each of the following
14146 headings:
- 14147 generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be supplied
14148 by the evaluation authority;
- 14149 bypassing;
- 14150 tampering;

- 14151 direct attacks;
- 14152 monitoring;
- 14153 misuse.
- 14154 Items b) - f) are explained in greater detail in Annex B.2.1.
- 14155 The security architecture description should be considered in light of each of the above generic
14156 potential vulnerabilities. Each potential vulnerability should be considered to search for possible
14157 ways in which to defeat the TSF protection and undermine the TSF.
- 14158 **Work unit AVA_VAN.5-5**
- 14159 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates
14160 for testing and applicable to the TOE in its operational environment.
- 14161 It may be identified that no further consideration of the potential vulnerability is required if for
14162 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
14163 prevent exploitation of the potential vulnerability in that operational environment. For instance,
14164 restricting physical access to the TOE to authorised users only may effectively render a potential
14165 vulnerability to tampering unexploitable.
- 14166 The evaluator records any reasons for exclusion of potential vulnerabilities from further
14167 consideration if the evaluator determines that the potential vulnerability is not applicable in the
14168 operational environment. Otherwise the evaluator records the potential vulnerability for further
14169 consideration.
- 14170 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
14171 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.
- 14172 **Action AVA_VAN.5.4E**
- 14173 **Work unit AVA_VAN.5-6**
- 14174 The evaluator ***shall devise*** penetration tests, based on the independent search for potential
14175 vulnerabilities.
- 14176 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
14177 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
14178 the sources of publicly available information and the analysis of the TOE guidance and design
14179 evidence. The evaluator should have access to current information (e.g. from the evaluation
14180 authority) regarding known potential vulnerabilities that may not have been considered by the
14181 evaluator.
- 14182 The evaluator is reminded that, as for considering the security architecture description in the
14183 search for vulnerabilities (as detailed in AVA_VAN.5-3), testing should be performed to confirm the
14184 architectural properties. If requirements from ATE_DPT are included in the SARs, the developer
14185 testing evidence will include testing performed to confirm the correct implementation of any
14186 specific mechanisms detailed in the security architecture description. However, the developer
14187 testing will not necessarily include testing of all aspects of the architectural properties that protect
14188 the TSF, as much of this testing will be negative testing in nature, attempting to disprove the
14189 properties. In developing the strategy for penetration testing, the evaluator will ensure that all
14190 aspects of the security architecture description are tested, either in functional testing (as
14191 considered in 15,) or evaluator penetration testing.
- 14192 The evaluator will probably find it practical to carry out penetration test using a series of test cases,
14193 where each test case will test for a specific potential vulnerability.

- 14194 The evaluator is not expected to test for potential vulnerabilities (including those in the public
14195 domain) beyond those which required a **High** attack potential. In some cases, however, it will be
14196 necessary to carry out a test before the exploitability can be determined. Where, as a result of
14197 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond **High**
14198 attack potential, this is reported in the ETR as a residual vulnerability.
- 14199 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be
14200 found in Annex B.4.
- 14201 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a **High** (or less)
14202 attack potential and resulting in a violation of the security objectives should be the highest priority
14203 potential vulnerabilities comprising the list used to direct penetration testing against the TOE.
- 14204 **Work unit AVA_VAN.5-7**
- 14205 The evaluator *shall produce* penetration test documentation for the tests based on the list of
14206 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
14207 documentation shall include:
- 14208 identification of the potential vulnerability the TOE is being tested for;
- 14209 instructions to connect and setup all required test equipment as required to conduct the
14210 penetration test;
- 14211 instructions to establish all penetration test prerequisite initial conditions;
- 14212 instructions to stimulate the TSF;
- 14213 instructions for observing the behaviour of the TSF;
- 14214 descriptions of all expected results and the necessary analysis to be performed on the observed
14215 behaviour for comparison against expected results;
- 14216 instructions to conclude the test and establish the necessary post-test state for the TOE.
- 14217 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
14218 identified during the search of the public domain and the analysis of the evaluation evidence.
- 14219 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
14220 those for which a **High** attack potential is required to effect an attack. However, as a result of
14221 evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only
14222 by an attacker with greater than **High** attack potential. Such vulnerabilities are to be reported in
14223 the ETR as residual vulnerabilities.
- 14224 With an understanding of the potential vulnerability, the evaluator determines the most feasible
14225 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 14226 the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses (It is
14227 possible that the evaluator will need to use an interface to the TOE other than the TSFI to
14228 demonstrate properties of the TSF such as those described in the security architecture description
14229 (as required by ADV_ARC). It should be noted, that although these TOE interfaces provide a means
14230 of testing the TSF properties, they are not the subject of the test.);
- 14231 initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will
14232 need to exist and security attributes they will need to have);
- 14233 special test equipment that will be required to either stimulate a TSFI or make observations of a
14234 TSFI;

- 14235 whether theoretical analysis should replace physical testing, particularly relevant where the
14236 results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are
14237 likely to succeed after a given number of attempts.
- 14238 The evaluator will probably find it practical to carry out penetration testing using a series of test
14239 cases, where each test case will test for a specific potential vulnerability.
- 14240 The intent of specifying this level of detail in the test documentation is to allow another evaluator
14241 to repeat the tests and obtain an equivalent result.
- 14242 **Work unit AVA_VAN.5-8**
- 14243 The evaluator ***shall conduct*** penetration testing.
- 14244 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.5-6 as a
14245 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
14246 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
14247 as a result of information learnt during penetration testing that, if performed by the evaluator, are
14248 to be recorded in the penetration test documentation. Such tests may be required to follow up
14249 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
14250 evaluator during the pre-planned testing.
- 14251 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the
14252 evaluator should determine whether or not the evaluator's own analysis was incorrect, or if
14253 evaluation deliverables are incorrect or incomplete.
- 14254 The evaluator is not expected to test for potential vulnerabilities (including those in the public
14255 domain) beyond those which required a **High** attack potential. In some cases, however, it will be
14256 necessary to carry out a test before the exploitability can be determined. Where, as a result of
14257 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond **High**
14258 attack potential, this is reported in the ETR as a residual vulnerability.
- 14259 **Work unit AVA_VAN.5-9**
- 14260 The evaluator ***shall record*** the actual results of the penetration tests.
- 14261 While some specific details of the actual test results may be different from those expected (e.g. time
14262 and date fields in an audit record) the overall result should be identical. Any unexpected test
14263 results should be investigated. The impact on the evaluation should be stated and justified.
- 14264 **Work unit AVA_VAN.5-10**
- 14265 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing
14266 approach, configuration, depth and results.
- 14267 The penetration testing information reported in the ETR allows the evaluator to convey the overall
14268 penetration testing approach and effort expended on this sub-activity. The intent of providing this
14269 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
14270 intended that the information regarding penetration testing in the ETR be an exact reproduction of
14271 specific test steps or results of individual penetration tests. The intention is to provide enough
14272 detail to allow other evaluators and evaluation authorities to gain some insight about the
14273 penetration testing approach chosen, amount of penetration testing performed, TOE test
14274 configurations, and the overall results of the penetration testing activity.
- 14275 Information that would typically be found in the ETR section regarding evaluator penetration
14276 testing efforts is:
- 14277 TOE test configurations. The particular configurations of the TOE that were penetration tested;

- 14278 TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of
14279 the penetration testing;
- 14280 Verdict for the sub-activity. The overall judgement on the results of penetration testing.
- 14281 This list is by no means exhaustive and is only intended to provide some context as to the type of
14282 information that should be present in the ETR concerning the penetration testing the evaluator
14283 performed during the evaluation.
- 14284
- 14285 **Work unit AVA_VAN.5-11**
- 14286 The evaluator **shall examine** the results of all penetration testing to determine that the TOE, in its
14287 operational environment, is resistant to an attacker possessing a **High** attack potential.
- 14288 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
14289 an attacker possessing an attack potential less than **or equal to** High, then this evaluator action
14290 fails.
- 14291 **This text was incorporated from a national scheme document (AIS34 from BSI). References within**
14292 **that text to other scheme documents (such as AIS14, 19, 26) have been deleted but additional text**
14293 **would be welcome where it might add to clarity**
- 14294 The guidance in B.4 and the guidance for special technical areas that is relevant for the national
14295 scheme should be used to determine the attack potential required to exploit a particular
14296 vulnerability and whether it can therefore be exploited in the intended environment. It may not be
14297 necessary for the attack potential to be calculated in every instance, only if there is some doubt as
14298 to whether or not the vulnerability can be exploited by an attacker possessing an attack potential
14299 less than **or equal to** High.
- 14300 **Work unit AVA_VAN.5-12**
- 14301 The evaluator **shall report** in the corresponding ETR-part all exploitable vulnerabilities and
14302 residual vulnerabilities, detailing for each:
- 14303 its source (e.g. ISO/IEC 18045 activity being undertaken when it was conceived, known to the
14304 evaluator, read in a publication);
- 14305 the SFR(s) not met;
- 14306 a description;
- 14307 whether it is exploitable in its operational environment or not (i.e. exploitable or residual);
- 14308 the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the
14309 equipment required to perform the identified vulnerabilities, and the corresponding values using
14310 the tables 3 and 4 of Annex B.4.
- 14311 Class ACO: Composition), the following will apply. In the instances that the component TOE under
14312 evaluation depends on other components in the operational environment to support their
14313 operation, the developer may wish to consider using the other component(s) that will be used in
14314 the composed TOE to fulfil the requirements of the operational environment as one of the test
14315 configurations. This will reduce the amount an additional testing that will be required for the
14316 composed TOE evaluation.

14317 **15.5.1.11.8 Work unit ATE_FUN.1-4**

14318 The evaluator **shall examine** the test plans to determine that sufficient instructions are provided
 14319 for any ordering dependencies.

14320 Some steps may have to be performed to establish initial conditions. For example, user accounts
 14321 need to be added before they can be deleted. An example of ordering dependencies on the results
 14322 of other tests is the need to perform actions in a test that will result in the generation of audit
 14323 records, before performing a test to consider the searching and sorting of those audit records.
 14324 Another example of an ordering dependency would be where one test case generates a file of data
 14325 to be used as input for another test case.

14326 The evaluator may wish to employ a sampling strategy when performing this work unit.

14327 ISO/IEC 15408-3 ATE_FUN.1.3C: *The expected test results shall show the anticipated outputs from a*
 14328 *successful execution of the tests.*

14329 **15.5.1.11.9 Work unit ATE_FUN.1-5**

14330 The evaluator **shall examine** the test documentation to determine that all expected tests results
 14331 are included.

14332 The expected test results are needed to determine whether or not a test has been successfully
 14333 performed. Expected test results are sufficient if they are unambiguous and consistent with
 14334 expected behaviour given the testing approach.

14335 The evaluator may wish to employ a sampling strategy when performing this work unit.

14336 ISO/IEC 15408-3 ATE_FUN.1.4C: *The actual test results shall be consistent with the expected test*
 14337 *results.*

14338 **15.5.1.11.10 Work unit ATE_FUN.1-6**

14339 The evaluator **shall check** that the actual test results in the test documentation are consistent with
 14340 the expected test results in the test documentation.

14341 A comparison of the actual and expected test results provided by the developer will reveal any
 14342 inconsistencies between the results. It may be that a direct comparison of actual results cannot be
 14343 made until some data reduction or synthesis has been first performed. In such cases, the
 14344 developer's test documentation should describe the process to reduce or synthesise the actual data.

14345 For example, the developer may need to test the contents of a message buffer after a network
 14346 connection has occurred to determine the contents of the buffer. The message buffer will contain a
 14347 binary number. This binary number would have to be converted to another form of data
 14348 representation in order to make the test more meaningful. The conversion of this binary
 14349 representation of data into a higher-level representation will have to be described by the developer
 14350 in enough detail to allow an evaluator to perform the conversion process (i.e. synchronous or
 14351 asynchronous transmission, number of stop bits, parity, etc.).

14352 It should be noted that the description of the process used to reduce or synthesise the actual data is
 14353 used by the evaluator not to actually perform the necessary modification but to assess whether this
 14354 process is correct. It is up to the developer to transform the expected test results into a format that
 14355 allows an easy comparison with the actual test results.

14356 The evaluator may wish to employ a sampling strategy when performing this work unit.

14357 **15.5.1.11.11 Work unit ATE_FUN.1-7**

14358 The evaluator ***shall report*** the developer testing effort, outlining the testing approach,
14359 configuration, depth and results.

14360 The developer testing information recorded in the ETR allows the evaluator to convey the overall
14361 testing approach and effort expended on the testing of the TOE by the developer. The intent of
14362 providing this information is to give a meaningful overview of the developer testing effort. It is not
14363 intended that the information regarding developer testing in the ETR be an exact reproduction of
14364 specific test steps or results of individual tests. The intention is to provide enough detail to allow
14365 other evaluators and evaluation authorities to gain some insight about the developer's testing
14366 approach, amount of testing performed, TOE test configurations, and the overall results of the
14367 developer testing.

14368 Information that would typically be found in the ETR subclause regarding the developer testing
14369 effort is:

14370 a) TOE test configurations. The particular configurations of the TOE that were tested,
14371 including whether any privileged code was required to set up the test or clean up
14372 afterwards;

14373 b) testing approach. An account of the overall developer testing strategy employed;

14374 c) testing results. A description of the overall developer testing results.

14375 This list is by no means exhaustive and is only intended to provide some context as to the type of
14376 information that should be present in the ETR concerning the developer testing effort.

14377 **15.5.2 Evaluation of sub-activity (ATE_FUN.2)**

14378 **15.5.2.1 Objectives**

14379 The objective of this sub-activity is to determine whether the developer correctly performed and
14380 documented the tests in the test documentation and to ensure that testing is structured such as to
14381 avoid circular arguments about the correctness of the interfaces being tested.

14382 **15.5.2.2 Input**

14383 The evaluation evidence for this sub-activity is:

14384 g) the ST;

14385 h) the functional specification;

14386 i) the test documentation.

14387 **15.5.2.3 Application notes**

14388 Although the test procedures may state pre-requisite initial test conditions in terms of ordering of
14389 tests, they may not provide a rationale for the ordering. An analysis of test ordering, which
14390 provides this rationale, is an important factor in determining the adequacy of testing, as there is a
14391 possibility of faults being concealed by the ordering of tests.

14392 **15.5.2.4 Action ATE_FUN.2.1E**

14393 ISO/IEC 15408-3 ATE_FUN.2.1C *The test documentation shall consist of test plans, expected test*
14394 *results and actual test results.*

14395 **15.5.2.4.1 Work unit ATE_FUN.2-1**

14396 The evaluator **shall check** that the test documentation includes test plans, expected test results and
 14397 actual test results.

14398 The evaluator checks that test plans, expected tests results and actual test results are included in
 14399 the test documentation.

14400 ISO/IEC 15408-3 ATE_FUN.2.2C *The test plans shall identify the tests to be performed and describe*
 14401 *the scenarios for performing each test. These scenarios shall include any ordering dependencies on the*
 14402 *results of other tests.*

14403 **15.5.2.4.2 Work unit ATE_FUN.2-2**

14404 The evaluator **shall examine** the test plan to determine that it describes the scenarios for
 14405 performing each test.

14406 The evaluator determines that the test plan provides information about the test configuration
 14407 being used: both on the configuration of the TOE and on any test equipment being used. This
 14408 information should be detailed enough to ensure that the test configuration is reproducible.

14409 The evaluator also determines that the test plan provides information about how to execute the
 14410 test: any necessary automated set-up procedures (and whether they require privilege to run),
 14411 inputs to be applied, how these inputs are applied, how output is obtained, any automated clean-up
 14412 procedures (and whether they require privilege to run), etc. This information should be detailed
 14413 enough to ensure that the test is reproducible.

14414 The evaluator may wish to employ a sampling strategy when performing this work unit.

14415 **15.5.2.4.3 Work unit ATE_FUN.2-3**

14416 The evaluator **shall examine** the test plan to determine that the TOE test configuration is
 14417 consistent with the ST.

14418 The TOE referred to in the developer's test plan should have the same unique reference as
 14419 established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

14420 It is possible for the ST to specify more than one configuration for evaluation. The evaluator
 14421 verifies that all test configurations identified in the developer test documentation are consistent
 14422 with the ST. For example, the ST might define configuration options that must be set, which could
 14423 have an impact upon what constitutes the TOE by including or excluding additional portions. The
 14424 evaluator verifies that all such variations of the TOE are considered.

14425 The evaluator should consider the security objectives for the operational environment described in
 14426 the ST that may apply to the test environment. There may be some objectives for the operational
 14427 environment that do not apply to the test environment. For example, an objective about user
 14428 clearances may not apply; however, an objective about a single point of connection to a network
 14429 would apply.

14430 The evaluator may wish to employ a sampling strategy when performing this work unit.

14431 If this work unit is applied to a component TOE that might be used/integrated in a composed TOE
 14432 (see Class ACO: Composition), the following will apply. In the instances that the component TOE
 14433 under evaluation depends on other components in the operational environment to support their
 14434 operation, the developer may wish to consider using the other component(s) that will be used in
 14435 the composed TOE to fulfil the requirements of the operational environment as one of the test
 14436 configurations. This will reduce the amount an additional testing that will be required for the
 14437 composed TOE evaluation.

14438 **15.5.2.4.4 Work unit ATE_FUN.2-4**

14439 The evaluator **shall examine** the test plans to determine that sufficient instructions are provided
14440 for any ordering dependencies.

14441 Some steps may have to be performed to establish initial conditions. For example, user accounts
14442 need to be added before they can be deleted. An example of ordering dependencies on the results
14443 of other tests is the need to perform actions in a test that will result in the generation of audit
14444 records, before performing a test to consider the searching and sorting of those audit records.
14445 Another example of an ordering dependency would be where one test case generates a file of data
14446 to be used as input for another test case.

14447 The evaluator may wish to employ a sampling strategy when performing this work unit.

14448 **ATE_FUN.2.3C** *The expected test results shall show the anticipated outputs from a*
14449 *successful execution of the tests.*

14450 **15.5.2.4.5 Work unit ATE_FUN.2-5**

14451 The evaluator **shall examine** the test documentation to determine that all expected tests results
14452 are included.

14453 The expected test results are needed to determine whether or not a test has been successfully
14454 performed. Expected test results are sufficient if they are unambiguous and consistent with
14455 expected behaviour given the testing approach.

14456 The evaluator may wish to employ a sampling strategy when performing this work unit.

14457 **ATE_FUN.2.4C** *The actual test results shall be consistent with the expected test results.*

14458 **15.5.2.4.6 Work unit ATE_FUN.2-6**

14459 The evaluator **shall check** that the actual test results in the test documentation are consistent with
14460 the expected test results in the test documentation.

14461 A comparison of the actual and expected test results provided by the developer will reveal any
14462 inconsistencies between the results. It may be that a direct comparison of actual results cannot be
14463 made until some data reduction or synthesis has been first performed. In such cases, the
14464 developer's test documentation should describe the process to reduce or synthesise the actual data.

14465 For example, the developer may need to test the contents of a message buffer after a network
14466 connection has occurred to determine the contents of the buffer. The message buffer will contain a
14467 binary number. This binary number would have to be converted to another form of data
14468 representation in order to make the test more meaningful. The conversion of this binary
14469 representation of data into a higher-level representation will have to be described by the developer
14470 in enough detail to allow an evaluator to perform the conversion process (i.e. synchronous or
14471 asynchronous transmission, number of stop bits, parity, etc.).

14472 It should be noted that the description of the process used to reduce or synthesise the actual data is
14473 used by the evaluator not to actually perform the necessary modification but to assess whether this
14474 process is correct. It is up to the developer to transform the expected test results into a format that
14475 allows an easy comparison with the actual test results.

14476 The evaluator may wish to employ a sampling strategy when performing this work unit.

14477 **15.5.2.4.7 Work unit ATE_FUN.2-7**

14478 The evaluator **shall report** the developer testing effort, outlining the testing approach,
14479 configuration, depth and results.

- 14480 The developer testing information recorded in the ETR allows the evaluator to convey the overall
 14481 testing approach and effort expended on the testing of the TOE by the developer. The intent of
 14482 providing this information is to give a meaningful overview of the developer testing effort. It is not
 14483 intended that the information regarding developer testing in the ETR be an exact reproduction of
 14484 specific test steps or results of individual tests. The intention is to provide enough detail to allow
 14485 other evaluators and evaluation authorities to gain some insight about the developer's testing
 14486 approach, amount of testing performed, TOE test configurations, and the overall results of the
 14487 developer testing.
- 14488 Information that would typically be found in the ETR section regarding the developer testing effort
 14489 is:
- 14490 g) TOE test configurations. The particular configurations of the TOE that were
 14491 tested, including whether any privileged code was required to set up the test or
 14492 clean up afterwards;
- 14493 h) testing approach. An account of the overall developer testing strategy
 14494 employed;
- 14495 i) testing results. A description of the overall developer testing results.
- 14496 This list is by no means exhaustive and is only intended to provide some context as to the type of
 14497 information that should be present in the ETR concerning the developer testing effort.
- 14498 **ATE_FUN.2.5C The test documentation shall include an analysis of the test**
 14499 **procedure ordering dependencies.**
- 14500 **15.5.2.4.8 Work unit ATE_FUN.2-8**
- 14501 The evaluator *shall examine* the analysis of the test procedure ordering dependencies to
 14502 determine that a sufficient justification for the chosen ordering of test cases is given.
- 14503 Usually the evaluator will generate a table of all cases, where the test documentation requires a
 14504 certain ordering of the tests and will then examine if sufficient justification is given in any case,
 14505 why testing in this ordering is adequate and sufficient.
- 14506 As an example we assume that the TSF provide a random number generator, which needs to be
 14507 initialised (for example with an adequate seed) before random numbers of a specified quality can
 14508 be generated. In this case the evaluator will consider the following question:
- 14509 Does the test documentation only describe an ordering of tests, where the initialisation is done
 14510 before calling the function to generate a random number?
- 14511 In this case the justification needs to show, why the developer expects, that in the intended
 14512 environment of the TOE the random number function will not be called without initialisation of the
 14513 random number generator.
- 14514 If for example the user guidance documentation includes a clear instruction that the random
 14515 number generator needs to be initialised adequately before being called, this may be considered
 14516 adequate as a justification. (note that the question if it can be plausibly assumed that users will
 14517 follow such instruction is covered by the evaluation activities for the classes ASE and AGD and
 14518 needs not to be re-examined here.)
- 14519 If, on the other hand, the TOE provides an authentication protocol, which implicitly uses random
 14520 numbers provided by the random number generator, and an attacker can therefore "call" the
 14521 random number generator implicitly by simply trying to authenticate himself, and if neither the
 14522 TOE nor the environment prevent an attacker from doing this even before the random number
 14523 generator is initialised, a test case needs to show, what happens in such situation.

14524 If, for example, instead of returning a "bad" random number, the random number function would
 14525 return an error, when called without proper initialisation, it would be much better to include a test
 14526 showing this secure behaviour instead of trying to justify why the functions are only tested in the
 14527 usual order.

14528 Note: Of course even without ATE_FUN.2 an evaluator would be expected to look for potential
 14529 vulnerabilities like the one described above. However, ATE_FUN.2.5C adds assurance by requiring
 14530 the developer to give a systematic justification, why their chosen order of test cases doesn't hide
 14531 such potential failures of security functions.

14532 **15.6 Independent testing (ATE_IND)**

14533 **15.6.1 Evaluation of sub-activity (ATE_IND.1)**

14534 **15.6.1.1 Objectives**

14535 The goal of this activity is to determine, by independently testing a subset of the TSFI, whether the
 14536 TOE behaves as specified in the functional specification and guidance documentation.

14537 **15.6.1.2 Input**

14538 The evaluation evidence for this sub-activity is:

- 14539 a) the ST;
- 14540 b) the functional specification;
- 14541 c) the operational user guidance;
- 14542 d) the preparative user guidance;
- 14543 e) the TOE suitable for testing.

14544 **15.6.1.3 Action ATE_IND.1.1E**

14545 ISO/IEC 15408-3 ATE_IND.1.1C: *The TOE shall be suitable for testing.*

14546 **15.6.1.3.1 Work unit ATE_IND.1-1**

14547 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
 14548 the configuration under evaluation as specified in the ST.

14549 The TOE provided by the developer should have the same unique reference as established by the
 14550 CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

14551 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
 14552 comprise a number of distinct hardware and software entities that need to be tested in accordance
 14553 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

14554 The evaluator should consider the security objectives for the operational environment described in
 14555 the ST that may apply to the test environment and ensure they are met in the testing environment.
 14556 There may be some objectives for the operational environment that do not apply to the test
 14557 environment. For example, an objective about user clearances may not apply; however, an
 14558 objective about a single point of connection to a network would apply.

14559 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
 14560 ensure that these resources are calibrated correctly.

14561 **15.6.1.3.2 Work unit ATE_IND.1-2**

14562 The evaluator *shall examine* the TOE to determine that it has been installed properly and is in a
14563 known state.

14564 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
14565 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
14566 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
14567 installed properly and is in a known state. If this is not the case, then the evaluator should follow
14568 the developer's procedures to install and start up the TOE, using the supplied guidance only.

14569 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
14570 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.

14571 **15.6.1.4 Action ATE_IND.1.2E**14572 **15.6.1.4.1 Work unit ATE_IND.1-3**

14573 The evaluator *shall devise* a test subset.

14574 The evaluator selects a test subset and testing strategy that is appropriate for the TOE. One
14575 extreme testing strategy would be to have the test subset contain as many interfaces as possible
14576 tested with little rigour. Another testing strategy would be to have the test subset contain a few
14577 interfaces based on their perceived relevance and rigorously test these interfaces.

14578 Typically the testing approach taken by the evaluator should fall somewhere between these two
14579 extremes. The evaluator should exercise most of the interfaces using at least one test, but testing
14580 need not demonstrate exhaustive specification testing.

14581 The evaluator, when selecting the subset of the interfaces to be tested, should consider the
14582 following factors:

14583 a) The number of interfaces from which to draw upon for the test subset. Where the TSF
14584 includes only a small number of relatively simple interfaces, it may be practical to
14585 rigorously test all of the interfaces. In other cases this may not be cost-effective, and
14586 sampling is required.

14587 b) Maintaining a balance of evaluation activities. The evaluator effort expended on the test
14588 activity should be commensurate with that expended on any other evaluation activity.

14589 The evaluator selects the interfaces to compose the subset. This selection will depend on a number
14590 of factors, and consideration of these factors may also influence the choice of test subset size:

14591 a) Significance of interfaces. Those interfaces more significant than others should be
14592 included in the test subset. One major factor of "significance" is the security-relevance
14593 (SFR-enforcing interfaces would be more significant than SFR-supporting interfaces,
14594 which are more significant than SFR-non-interfering interfaces; see ISO/IEC 15408-3
14595 Subclause Functional specification (ADV_FSP)). The other major factor of "significance" is
14596 the number of SFRs mapping to this interface (as determined when identifying the
14597 correspondence between levels of abstraction in ADV).

14598 b) Complexity of the interface. Complex interfaces may require complex tests that impose
14599 onerous requirements on the developer or evaluator, which may not be conducive to
14600 cost-effective evaluations. Conversely, they are a likely area to find errors and are good
14601 candidates for the subset. The evaluator will need to strike a balance between these
14602 considerations.

- 14603 c) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and
14604 their inclusion in the subset may maximise the number of interfaces tested (albeit
14605 implicitly). Certain interfaces will typically be used to provide a variety of security
14606 functionality, and will tend to be the target of an effective testing approach.
- 14607 d) Types of interfaces (e.g. programmatic, command-line, protocol). The evaluator should
14608 consider including tests for all different types of interfaces that the TOE supports.
- 14609 e) Interfaces that give rise to features that are innovative or unusual. Where the TOE
14610 contains innovative or unusual features, which may feature strongly in marketing
14611 literature and guidance documents, the corresponding interfaces should be strong
14612 candidates for testing.
- 14613 This guidance articulates factors to consider during the selection process of an appropriate test
14614 subset, but these are by no means exhaustive.
- 14615 **15.6.1.4.2 Work unit ATE_IND.1-4**
- 14616 The evaluator **shall produce** test documentation for the test subset that is sufficiently detailed to
14617 enable the tests to be reproducible.
- 14618 With an understanding of the expected behaviour of the TSF, from the ST and the functional
14619 specification, the evaluator has to determine the most feasible way to test the interface. Specifically
14620 the evaluator considers:
- 14621 a) the approach that will be used, for instance, whether an external interface will be tested,
14622 or an internal interface using a test harness, or will an alternate test approach be
14623 employed (e.g. in exceptional circumstances, a code inspection, if the implementation
14624 representation is available);
- 14625 b) the interface(s) that will be used to test and observe responses;
- 14626 c) the initial conditions that will need to exist for the test (i.e. any particular objects or
14627 subjects that will need to exist and security attributes they will need to have);
- 14628 d) special test equipment that will be required to either stimulate an interface (e.g. packet
14629 generators) or make observations of an interface (e.g. network analysers).
- 14630 The evaluator may find it practical to test each interface using a series of test cases, where each test
14631 case will test a very specific aspect of expected behaviour.
- 14632 The evaluator's test documentation should specify the derivation of each test, tracing it back to the
14633 relevant interface(s).
- 14634 **15.6.1.4.3 Work unit ATE_IND.1-5**
- 14635 The evaluator **shall conduct** testing.
- 14636 The evaluator uses the test documentation developed as a basis for executing tests on the TOE. The
14637 test documentation is used as a basis for testing but this does not preclude the evaluator from
14638 performing additional ad hoc tests. The evaluator may devise new tests based on behaviour of the
14639 TOE discovered during testing. These new tests are recorded in the test documentation.
- 14640 **15.6.1.4.4 Work unit ATE_IND.1-6**
- 14641 The evaluator **shall record** the following information about the tests that compose the test subset:
- 14642 a) identification of the interface behaviour to be tested;

- 14643 b) instructions to connect and setup all required test equipment as required to conduct the
14644 test;
- 14645 c) instructions to establish all prerequisite test conditions;
- 14646 d) instructions to stimulate the interface;
- 14647 e) instructions for observing the behaviour of the interface;
- 14648 f) descriptions of all expected results and the necessary analysis to be performed on the
14649 observed behaviour for comparison against expected results;
- 14650 g) instructions to conclude the test and establish the necessary post-test state for the TOE;
- 14651 h) actual test results.
- 14652 The level of detail should be such that another evaluator could repeat the tests and obtain an
14653 equivalent result. While some specific details of the test results may be different (e.g. time and date
14654 fields in an audit record) the overall result should be identical.
- 14655 There may be instances when it is unnecessary to provide all the information presented in this
14656 work unit (e.g. the actual test results of a test may not require any analysis before a comparison
14657 between the expected results can be made). The determination to omit this information is left to
14658 the evaluator, as is the justification.
- 14659 **15.6.1.4.5 Work unit ATE_IND.1-7**
- 14660 The evaluator **shall check** that all actual test results are consistent with the expected test results.
- 14661 Any differences in the actual and expected test results may indicate that the TOE does not perform
14662 as specified or that the evaluator test documentation may be incorrect. Unexpected actual results
14663 may require corrective maintenance to the TOE or test documentation and perhaps require re-
14664 running of impacted tests and modifying the test sample size and composition. This determination
14665 is left to the evaluator, as is its justification.
- 14666 **15.6.1.4.6 Work unit ATE_IND.1-8**
- 14667 The evaluator **shall report** in the ETR the evaluator testing effort, outlining the testing approach,
14668 configuration, depth and results.
- 14669 The evaluator testing information reported in the ETR allows the evaluator to convey the overall
14670 testing approach and effort expended on the testing activity during the evaluation. The intent of
14671 providing this information is to give a meaningful overview of the testing effort. It is not intended
14672 that the information regarding testing in the ETR be an exact reproduction of specific test
14673 instructions or results of individual tests. The intention is to provide enough detail to allow other
14674 evaluators and evaluation authorities to gain some insight about the testing approach chosen,
14675 amount of testing performed, TOE test configurations, and the overall results of the testing activity.
- 14676 Information that would typically be found in the ETR subclause regarding the evaluator testing
14677 effort is:
- 14678 a) TOE test configurations. The particular configurations of the TOE that were tested;
- 14679 b) subset size chosen. The amount of interfaces that were tested during the evaluation and a
14680 justification for the size;
- 14681 c) selection criteria for the interfaces that compose the subset. Brief statements about the
14682 factors considered when selecting interfaces for inclusion in the subset;

- 14683 d) interfaces tested. A brief listing of the interfaces that merited inclusion in the subset;
- 14684 e) verdict for the activity. The overall judgement on the results of testing during the
14685 evaluation.
- 14686 This list is by no means exhaustive and is only intended to provide some context as to the type of
14687 information that should be present in the ETR concerning the testing the evaluator performed
14688 during the evaluation.

14689 **15.6.2 Evaluation of sub-activity (ATE_IND.2)**

14690 **15.6.2.1 Objectives**

14691 The goal of this activity is to determine, by independently testing a subset of the TSF, whether the
14692 TOE behaves as specified in the design documentation, and to gain confidence in the developer's
14693 test results by performing a sample of the developer's tests.

14694 **15.6.2.2 Input**

14695 The evaluation evidence for this sub-activity is:

- 14696 a) the ST;
- 14697 b) the functional specification;
- 14698 c) the TOE design description;
- 14699 d) the operational user guidance;
- 14700 e) the preparative user guidance;
- 14701 f) the configuration management documentation;
- 14702 g) the test documentation;
- 14703 h) the TOE suitable for testing.

14704 **15.6.2.3 Action ATE_IND.2.1E**

14705 ISO/IEC 15408-3 ATE_IND.2.1C: *The TOE shall be suitable for testing.*

14706 **15.6.2.3.1 Work unit ATE_IND.2-1**

14707 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
14708 the configuration under evaluation as specified in the ST.

14709 The TOE provided by the developer and identified in the test plan should have the same unique
14710 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
14711 introduction.

14712 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
14713 comprise a number of distinct hardware and software entities that need to be tested in accordance
14714 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

14715 The evaluator should consider the security objectives for the operational environment described in
14716 the ST that may apply to the test environment and ensure they are met in the testing environment.
14717 There may be some objectives for the operational environment that do not apply to the test

- 14718 environment. For example, an objective about user clearances may not apply; however, an
14719 objective about a single point of connection to a network would apply.
- 14720 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
14721 ensure that these resources are calibrated correctly.
- 14722 **15.6.2.3.2 Work unit ATE_IND.2-2**
- 14723 The evaluator **shall examine** the TOE to determine that it has been installed properly and is in a
14724 known state.
- 14725 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
14726 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
14727 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
14728 installed properly and is in a known state. If this is not the case, then the evaluator should follow
14729 the developer's procedures to install and start up the TOE, using the supplied guidance only.
- 14730 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
14731 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.
- 14732 ISO/IEC 15408-3 ATE_IND.2.2C: *The developer shall provide an equivalent set of resources to those*
14733 *that were used in the developer's functional testing of the TSF.*
- 14734 **15.6.2.3.3 Work unit ATE_IND.2-3**
- 14735 The evaluator **shall examine** the set of resources provided by the developer to determine that they
14736 are equivalent to the set of resources used by the developer to functionally test the TSF.
- 14737 The set of resource used by the developer is documented in the developer test plan, as considered
14738 in the Functional tests (ATE_FUN) family. The resource set may include laboratory access and
14739 special test equipment, among others. Resources that are not identical to those used by the
14740 developer need to be equivalent in terms of any impact they may have on test results.
- 14741 **15.6.2.4 Action ATE_IND.2.2E**
- 14742 **15.6.2.4.1 Work unit ATE_IND.2-4**
- 14743 The evaluator **shall conduct** testing using a sample of tests found in the developer test plan and
14744 procedures.
- 14745 The overall aim of this work unit is to perform a sufficient number of the developer tests to confirm
14746 the validity of the developer's test results. The evaluator has to decide on the size of the sample,
14747 and the developer tests that will compose the sample (see A.2).
- 14748 All the developer tests can be traced back to specific interfaces. Therefore, the factors to consider
14749 in the selection of the tests to compose the sample are similar to those listed for subset selection in
14750 work-unit ATE_IND.2-6. Additionally, the evaluator may wish to employ a random sampling
14751 method to select developer tests to include in the sample.
- 14752 **15.6.2.4.2 Work unit ATE_IND.2-5**
- 14753 The evaluator **shall check** that all the actual test results are consistent with the expected test
14754 results.
- 14755 Inconsistencies between the developer's expected test results and actual test results will compel
14756 the evaluator to resolve the discrepancies. Inconsistencies encountered by the evaluator could be
14757 resolved by a valid explanation and resolution of the inconsistencies by the developer.

14758 If a satisfactory explanation or resolution can not be reached, the evaluator's confidence in the
 14759 developer's test results may be lessened and it may be necessary for the evaluator to increase the
 14760 sample size to the extent that the subset identified in work unit ATE_IND.2-4 is adequately tested:
 14761 deficiencies with the developer's tests need to result in either corrective action to the TOE by the
 14762 developer (e.g., if the inconsistency is caused by incorrect behaviour) or to the developer's tests
 14763 (e.g., if the inconsistency is caused by an incorrect test), or in the production of new tests by the
 14764 evaluator.

14765 **15.6.2.5 Action ATE_IND.2.3E**

14766 **15.6.2.5.1 Work unit ATE_IND.2-6**

14767 The evaluator *shall devise* a test subset.

14768 The evaluator selects a test subset and testing strategy that is appropriate for the TOE. One
 14769 extreme testing strategy would be to have the test subset contain as many interfaces as possible
 14770 tested with little rigour. Another testing strategy would be to have the test subset contain a few
 14771 interfaces based on their perceived relevance and rigorously test these interfaces.

14772 Typically the testing approach taken by the evaluator should fall somewhere between these two
 14773 extremes. The evaluator should exercise most of the interfaces using at least one test, but testing
 14774 need not demonstrate exhaustive specification testing.

14775 The evaluator, when selecting the subset of the interfaces to be tested, should consider the
 14776 following factors:

- 14777 a) The developer test evidence. The developer test evidence consists of: the test
 14778 documentation, the available test coverage analysis, and the available depth of testing
 14779 analysis. The developer test evidence will provide insight as to how the TSF has been
 14780 exercised by the developer during testing. The evaluator applies this information when
 14781 developing new tests to independently test the TOE. Specifically the evaluator should
 14782 consider:
 - 14783 1) augmentation of developer testing for interfaces. The evaluator may wish to perform
 14784 more of the same type of tests by varying parameters to more rigorously test the interface.
 - 14785 2) supplementation of developer testing strategy for interfaces. The evaluator may wish to
 14786 vary the testing approach of a specific interface by testing it using another test strategy.
 - 14787 b) The number of interfaces from which to draw upon for the test subset. Where the TSF
 14788 includes only a small number of relatively simple interfaces, it may be practical to
 14789 rigorously test all of them. In other cases this may not be cost-effective, and sampling is
 14790 required.
 - 14791 c) Maintaining a balance of evaluation activities. The evaluator effort expended on the test
 14792 activity should be commensurate with that expended on any other evaluation activity.
- 14793 The evaluator selects the interfaces to compose the subset. This selection will depend on a number
 14794 of factors, and consideration of these factors may also influence the choice of test subset size:
- 14795 a) Rigour of developer testing of the interfaces. Those interfaces that the evaluator
 14796 determines require additional testing should be included in the test subset.
 - 14797 b) Developer test results. If the results of developer tests cause the evaluator to doubt that
 14798 an interface is not properly implemented, then the evaluator should include such
 14799 interfaces in the test subset.

- 14800 c) Significance of interfaces. Those interfaces more significant than others should be
 14801 included in the test subset. One major factor of “significance” is the security-relevance
 14802 (SFR-enforcing interfaces would be more significant than SFR-supporting interfaces,
 14803 which are more significant than SFR-non-interfering interfaces; see ISO/IEC 15408-3
 14804 Subclause ADV_FSP). The other major factor of “significance” is the number of SFRs
 14805 mapping to this interface (as determined when identifying the correspondence between
 14806 levels of abstraction in ADV).
- 14807 d) Complexity of interfaces. Interfaces that require complex implementation may require
 14808 complex tests that impose onerous requirements on the developer or evaluator, which
 14809 may not be conducive to cost-effective evaluations. Conversely, they are a likely area to
 14810 find errors and are good candidates for the subset. The evaluator will need to strike a
 14811 balance between these considerations.
- 14812 e) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and
 14813 their inclusion in the subset may maximise the number of interfaces tested (albeit
 14814 implicitly). Certain interfaces will typically be used to provide a variety of security
 14815 functionality, and will tend to be the target of an effective testing approach.
- 14816 f) Types of interfaces (e.g. programmatic, command-line, protocol). The evaluator should
 14817 consider including tests for all different types of interfaces that the TOE supports.
- 14818 g) Interfaces that give rise to features that are innovative or unusual. Where the TOE
 14819 contains innovative or unusual features, which may feature strongly in marketing
 14820 literature and guidance documents, the corresponding interfaces should be strong
 14821 candidates for testing.
- 14822 This guidance articulates factors to consider during the selection process of an appropriate test
 14823 subset, but these are by no means exhaustive.
- 14824 **15.6.2.5.2 Work unit ATE_IND.2-7**
- 14825 The evaluator ***shall produce*** test documentation for the test subset that is sufficiently detailed to
 14826 enable the tests to be reproducible.
- 14827 With an understanding of the expected behaviour of the TSF, from the ST, the functional
 14828 specification, and the TOE design description, the evaluator has to determine the most feasible way
 14829 to test the interface. Specifically the evaluator considers:
- 14830 a) the approach that will be used, for instance, whether an external interface will be tested,
 14831 or an internal interface using a test harness, or will an alternate test approach be
 14832 employed (e.g. in exceptional circumstances, a code inspection);
- 14833 b) the interface(s) that will be used to test and observe responses;
- 14834 c) the initial conditions that will need to exist for the test (i.e. any particular objects or
 14835 subjects that will need to exist and security attributes they will need to have);
- 14836 d) special test equipment that will be required to either stimulate an interface (e.g. packet
 14837 generators) or make observations of an interface (e.g. network analysers).
- 14838 The evaluator may find it practical to test each interface using a series of test cases, where each test
 14839 case will test a very specific aspect of expected behaviour of that interface.
- 14840 The evaluator’s test documentation should specify the derivation of each test, tracing it back to the
 14841 relevant interface(s).

14842 **15.6.2.5.3 Work unit ATE_IND.2-8**

14843 The evaluator **shall conduct** testing.

14844 The evaluator uses the test documentation developed as a basis for executing tests on the TOE. The
14845 test documentation is used as a basis for testing but this does not preclude the evaluator from
14846 performing additional ad hoc tests. The evaluator may devise new tests based on behaviour of the
14847 TOE discovered during testing. These new tests are recorded in the test documentation.

14848 **15.6.2.5.4 Work unit ATE_IND.2-9**

14849 The evaluator **shall record** the following information about the tests that compose the test subset:

14850 a) identification of the interface behaviour to be tested;

14851 b) instructions to connect and setup all required test equipment as required to conduct the
14852 test;

14853 c) instructions to establish all prerequisite test conditions;

14854 d) instructions to stimulate the interface;

14855 e) instructions for observing the interface;

14856 f) descriptions of all expected results and the necessary analysis to be performed on the
14857 observed behaviour for comparison against expected results;

14858 g) instructions to conclude the test and establish the necessary post-test state for the TOE;

14859 h) actual test results.

14860 The level of detail should be such that another evaluator could repeat the tests and obtain an
14861 equivalent result. While some specific details of the test results may be different (e.g. time and date
14862 fields in an audit record) the overall result should be identical.

14863 There may be instances when it is unnecessary to provide all the information presented in this
14864 work unit (e.g. the actual test results of a test may not require any analysis before a comparison
14865 between the expected results can be made). The determination to omit this information is left to
14866 the evaluator, as is the justification.

14867 **15.6.2.5.5 Work unit ATE_IND.2-10**

14868 The evaluator **shall check** that all actual test results are consistent with the expected test results.

14869 Any differences in the actual and expected test results may indicate that the TOE does not perform
14870 as specified or that the evaluator test documentation may be incorrect. Unexpected actual results
14871 may require corrective maintenance to the TOE or test documentation and perhaps require re-
14872 running of impacted tests and modifying the test sample size and composition. This determination
14873 is left to the evaluator, as is its justification.

14874 **15.6.2.5.6 Work unit ATE_IND.2-11**

14875 The evaluator **shall report** in the ETR the evaluator testing effort, outlining the testing approach,
14876 configuration, depth and results.

14877 The evaluator testing information reported in the ETR allows the evaluator to convey the overall
14878 testing approach and effort expended on the testing activity during the evaluation. The intent of
14879 providing this information is to give a meaningful overview of the testing effort. It is not intended

14880 that the information regarding testing in the ETR be an exact reproduction of specific test
 14881 instructions or results of individual tests. The intention is to provide enough detail to allow other
 14882 evaluators and evaluation authorities to gain some insight about the testing approach chosen,
 14883 amount of evaluator testing performed, amount of developer tests performed, TOE test
 14884 configurations, and the overall results of the testing activity.

14885 Information that would typically be found in the ETR subclause regarding the evaluator testing
 14886 effort is:

- 14887 a) TOE test configurations. The particular configurations of the TOE that were tested.
- 14888 b) subset size chosen. The amount of interfaces that were tested during the evaluation and a
 14889 justification for the size.
- 14890 c) selection criteria for the interfaces that compose the subset. Brief statements about the
 14891 factors considered when selecting interfaces for inclusion in the subset.
- 14892 d) Interfaces tested. A brief listing of the interfaces that merited inclusion in the subset.
- 14893 e) developer tests performed. The amount of developer tests performed and a brief
 14894 description of the criteria used to select the tests.
- 14895 f) verdict for the activity. The overall judgement on the results of testing during the
 14896 evaluation.

14897 This list is by no means exhaustive and is only intended to provide some context as to the type of
 14898 information that should be present in the ETR concerning the testing the evaluator performed
 14899 during the evaluation.

14900 **15.6.3 Evaluation of sub-activity (ATE_IND.3)**

14901 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

14902 **15.7 [PLACE-HOLDER] TOE Modular Testing Knowledge (ATE_MTK)**

14903 If the modularity approach included in ASE_AMA, ADV_MTC, ATE_MTK, ATE_MTT remains in
 14904 ISO/IEC 15408-x then work units will be required to cover these.

14905 **15.8** Suggestions for text would be welcomed in response to CD1 review. **If none are**
 14906 **received then this topic will be left to the next revision. [PLACE-HOLDER] TOE**
 14907 **Modular Traceability of Functional Requirements in Tests (ATE_MTT)**

14908 If the modularity approach included in ASE_AMA, ADV_MTC, ATE_MTK, ATE_MTT remains in
 14909 ISO/IEC 15408-x then work units will be required to cover these.

14910 **Suggestions for text would be welcomed in response to CD1 review.** If none are received then
 14911 this topic will be left to the next revision.

14912 **16 Class AVA: Vulnerability assessment**

14913 **16.1 Introduction**

14914 The purpose of the vulnerability assessment activity is to determine the exploitability of flaws or
 14915 weaknesses in the TOE in the operational environment. This determination is based upon analysis
 14916 of the evaluation evidence and a search of publicly available material by the evaluator and is
 14917 supported by evaluator penetration testing.

14918 **16.2 Vulnerability analysis (AVA_VAN)**

14919 **16.2.1 Evaluation of sub-activity (AVA_VAN.1)**

14920 **16.2.1.1 Objectives**

14921 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
14922 has easily identifiable exploitable vulnerabilities.

14923 **16.2.1.2 Input**

14924 The evaluation evidence for this sub-activity is:

- 14925 a) the ST;
- 14926 b) the guidance documentation;
- 14927 c) the TOE suitable for testing;
- 14928 d) information publicly available to support the identification of potential vulnerabilities.

14929 Other input for this sub-activity is:

- 14930 a) current information regarding potential vulnerabilities (e.g. from an evaluation authority).

14931 **16.2.1.3 Application notes**

14932 The evaluator should consider performing additional tests as a result of potential vulnerabilities
14933 encountered during the conduct of other parts of the evaluation.

14934 The use of the term guidance in this sub-activity refers to the operational guidance and the
14935 preparative guidance.

14936 Potential vulnerabilities may be in information that is publicly available, or not, and may require
14937 skill to exploit, or not. These two aspects are related, but are distinct. It should not be assumed that,
14938 simply because a potential vulnerability is identifiable from information that is publicly available, it
14939 can be easily exploited.

14940 **16.2.1.4 Action AVA_VAN.1.1E**

14941 ISO/IEC 15408-3 AVA_VAN.1.1C: *The TOE shall be suitable for testing.*

14942 **16.2.1.4.1 Work unit AVA_VAN.1-1**

14943 The evaluator **shall examine** the TOE to determine that the test configuration is consistent with
14944 the configuration under evaluation as specified in the ST.

14945 The TOE provided by the developer and identified in the test plan should have the same unique
14946 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
14947 introduction.

14948 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
14949 comprise a number of distinct hardware and software entities that need to be tested in accordance
14950 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

14951 The evaluator should consider the security objectives for the operational environment described in
14952 the ST that may apply to the test environment and ensure they are met in the testing environment.
14953 There may be some objectives for the operational environment that do not apply to the test

- 14954 environment. For example, an objective about user clearances may not apply; however, an
14955 objective about a single point of connection to a network would apply.
- 14956 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
14957 ensure that these resources are calibrated correctly.
- 14958 **16.2.1.4.2 Work unit AVA_VAN.1-2**
- 14959 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a
14960 known state
- 14961 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
14962 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
14963 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
14964 installed properly and is in a known state. If this is not the case, then the evaluator should follow
14965 the developer's procedures to install and start up the TOE, using the supplied guidance only.
- 14966 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
14967 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.
- 14968 **16.2.1.5 Action AVA_VAN.1.2E**
- 14969 **16.2.1.5.1 Work unit AVA_VAN.1-3**
- 14970 The evaluator ***shall examine*** sources of information publicly available to identify potential
14971 vulnerabilities in the TOE.
- 14972 The evaluator examines the sources of information publicly available to support the identification
14973 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
14974 information, which should be considered, e.g. mailing lists and security forums on the world wide
14975 web that report known vulnerabilities in specified technologies.
- 14976 The evaluator should not constrain their consideration of publicly available information to the
14977 above, but should consider any other relevant information available.
- 14978 While examining the evidence provided the evaluator will use the information in the public domain
14979 to further search for potential vulnerabilities. Where the evaluators have identified areas of
14980 concern, the evaluator should consider information publicly available that relate to those areas of
14981 concern.
- 14982 The availability of information that may be readily available to an attacker that helps to identify
14983 and facilitate attacks effectively operates to substantially enhance the attack potential of a given
14984 attacker. The accessibility of vulnerability information and sophisticated attack tools on the
14985 Internet makes it more likely that this information will be used in attempts to identify potential
14986 vulnerabilities in the TOE and exploit them. Modern search tools make such information easily
14987 available to the evaluator, and the determination of resistance to published potential
14988 vulnerabilities and well known generic attacks can be achieved in a cost-effective manner.
- 14989 The search of the information publicly available should be focused on those sources that refer
14990 specifically to the product from which the TOE is derived. The extensiveness of this search should
14991 consider the following factors: TOE type, evaluator experience in this TOE type, expected attack
14992 potential and the level of ADV evidence available.
- 14993 The identification process is iterative, where the identification of one potential vulnerability may
14994 lead to identifying another area of concern that requires further investigation.
- 14995 The evaluator will report what actions were taken to identify potential vulnerabilities in the
14996 information publicly available. However, in this type of search, the evaluator may not be able to

- 14997 describe the steps in identifying potential vulnerabilities before the outset of the examination, as
14998 the approach may evolve as a result of findings during the search.
- 14999 The evaluator will report the evidence examined in completing the search for potential
15000 vulnerabilities.
- 15001 **16.2.1.5.2 Work unit AVA_VAN.1-4**
- 15002 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates
15003 for testing and applicable to the TOE in its operational environment.
- 15004 It may be identified that no further consideration of the potential vulnerability is required if for
15005 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
15006 prevent exploitation of the potential vulnerability in that operational environment. For instance,
15007 restricting physical access to the TOE to authorised users only may effectively render a potential
15008 vulnerability to tampering unexploitable.
- 15009 The evaluator records any reasons for exclusion of potential vulnerabilities from further
15010 consideration if the evaluator determines that the potential vulnerability is not applicable in the
15011 operational environment. Otherwise the evaluator records the potential vulnerability for further
15012 consideration.
- 15013 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
15014 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.
- 15015 **16.2.1.6 Action AVA_VAN.1.3E**
- 15016 **16.2.1.6.1 Work unit AVA_VAN.1-5**
- 15017 The evaluator ***shall devise*** penetration tests, based on the independent search for potential
15018 vulnerabilities.
- 15019 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
15020 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
15021 the sources of information publicly available. Any current information provided to the evaluator by
15022 a third party (e.g. evaluation authority) regarding known potential vulnerabilities will be
15023 considered by the evaluator, together with any encountered potential vulnerabilities resulting
15024 from the performance of other evaluation activities.
- 15025 The evaluator will probably find it practical to carry out penetration test using a series of test cases,
15026 where each test case will test for a specific potential vulnerability.
- 15027 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15028 domain) beyond those which required a Basic attack potential. In some cases, however, it will be
15029 necessary to carry out a test before the exploitability can be determined. Where, as a result of
15030 evaluation expertise, the evaluator discovers a potential vulnerability that is beyond Basic attack
15031 potential, this is reported in the ETR as a residual vulnerability.
- 15032 **16.2.1.6.2 Work unit AVA_VAN.1-6**
- 15033 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of
15034 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
15035 documentation shall include:
- 15036 a) identification of the potential vulnerability the TOE is being tested for;
- 15037 b) instructions to connect and setup all required test equipment as required to conduct the
15038 penetration test;

- 15039 c) instructions to establish all penetration test prerequisite initial conditions;
- 15040 d) instructions to stimulate the TSF;
- 15041 e) instructions for observing the behaviour of the TSF;
- 15042 f) descriptions of all expected results and the necessary analysis to be performed on the
15043 observed behaviour for comparison against expected results;
- 15044 g) instructions to conclude the test and establish the necessary post-test state for the TOE.
- 15045 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
15046 identified during the search of the public domain.
- 15047 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
15048 those for which a Basic attack potential is required to effect an attack. However, as a result of
15049 evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only
15050 by an attacker with greater than Basic attack potential. Such vulnerabilities are to be reported in
15051 the ETR as residual vulnerabilities.
- 15052 With an understanding of the potential vulnerability, the evaluator determines the most feasible
15053 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 15054 a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe
15055 responses;
- 15056 b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects
15057 that will need to exist and security attributes they will need to have);
- 15058 c) special test equipment that will be required to either stimulate a TSFI or make
15059 observations of a TSFI (although it is unlikely that specialist equipment would be
15060 required to exploit a potential vulnerability assuming a Basic attack potential);
- 15061 d) whether theoretical analysis should replace physical testing, particularly relevant where
15062 the results of an initial test can be extrapolated to demonstrate that repeated attempts of
15063 an attack are likely to succeed after a given number of attempts.
- 15064 The evaluator will probably find it practical to carry out penetration testing using a series of test
15065 cases, where each test case will test for a specific potential vulnerability.
- 15066 The intent of specifying this level of detail in the test documentation is to allow another evaluator
15067 to repeat the tests and obtain an equivalent result.
- 15068 **16.2.1.6.3 Work unit AVA_VAN.1-7**
- 15069 The evaluator ***shall conduct*** penetration testing.
- 15070 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.1-5 as a
15071 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
15072 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
15073 as a result of information learnt during penetration testing that, if performed by the evaluator, are
15074 to be recorded in the penetration test documentation. Such tests may be required to follow up
15075 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
15076 evaluator during the pre-planned testing.
- 15077 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15078 domain) beyond those which required a Basic attack potential. In some cases, however, it will be
15079 necessary to carry out a test before the exploitability can be determined. Where, as a result of

15080 evaluation expertise, the evaluator discovers a potential vulnerability that is beyond Basic attack
15081 potential, this is reported in the ETR as a residual vulnerability.

15082 **16.2.1.6.4 Work unit AVA_VAN.1-8**

15083 The evaluator **shall record** the actual results of the penetration tests.

15084 While some specific details of the actual test results may be different from those expected (e.g. time
15085 and date fields in an audit record) the overall result should be identical. Any unexpected test
15086 results should be investigated. The impact on the evaluation should be stated and justified.

15087 **16.2.1.6.5 Work unit AVA_VAN.1-9**

15088 The evaluator **shall report** in the ETR the evaluator penetration testing effort, outlining the testing
15089 approach, configuration, depth and results.

15090 The penetration testing information reported in the ETR allows the evaluator to convey the overall
15091 penetration testing approach and effort expended on this sub-activity. The intent of providing this
15092 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
15093 intended that the information regarding penetration testing in the ETR be an exact reproduction of
15094 specific test steps or results of individual penetration tests. The intention is to provide enough
15095 detail to allow other evaluators and evaluation authorities to gain some insight about the
15096 penetration testing approach chosen, amount of penetration testing performed, TOE test
15097 configurations, and the overall results of the penetration testing activity.

15098 Information that would typically be found in the ETR subclause regarding evaluator penetration
15099 testing efforts is:

15100 a) TOE test configurations. The particular configurations of the TOE that were penetration
15101 tested;

15102 b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the
15103 focus of the penetration testing;

15104 c) verdict for the sub-activity. The overall judgement on the results of penetration testing.

15105 This list is by no means exhaustive and is only intended to provide some context as to the type of
15106 information that should be present in the ETR concerning the penetration testing the evaluator
15107 performed during the evaluation.

15108 **16.2.1.6.6 Work unit AVA_VAN.1-10**

15109 The evaluator **shall examine** the results of all penetration testing to determine that the TOE, in its
15110 operational environment, is resistant to an attacker possessing a Basic attack potential.

15111 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
15112 an attacker possessing less than Enhanced-Basic attack potential, then this evaluator action fails.

15113 The guidance in B.4 should be used to determine the attack potential required to exploit a
15114 particular vulnerability and whether it can therefore be exploited in the intended environment. It
15115 may not be necessary for the attack potential to be calculated in every instance, only if there is
15116 some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an
15117 attack potential less than Enhanced-Basic.

15118 **16.2.1.6.7 Work unit AVA_VAN.1-11**

15119 The evaluator **shall report** in the ETR all exploitable vulnerabilities and residual vulnerabilities,
15120 detailing for each:

- 15121 a) its source (e.g. evaluation methodology activity being undertaken when it was conceived,
15122 known to the evaluator, read in a publication);
- 15123 b) the SFR(s) not met;
- 15124 c) a description;
- 15125 d) whether it is exploitable in its operational environment or not (i.e. exploitable or
15126 residual).
- 15127 e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity
15128 and the equipment required to perform the identified vulnerabilities, and the
15129 corresponding values using the tables B.2 and B.3 of Annex B.4.

15130 **16.2.2 Evaluation of sub-activity (AVA_VAN.2)**

15131 **16.2.2.1 Objectives**

15132 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
15133 has vulnerabilities exploitable by attackers possessing Basic attack potential.

15134 **16.2.2.2 Input**

15135 The evaluation evidence for this sub-activity is:

- 15136 a) the ST;
- 15137 b) the functional specification;
- 15138 c) the TOE design;
- 15139 d) the security architecture description;
- 15140 e) the guidance documentation;
- 15141 f) the TOE suitable for testing;
- 15142 g) information publicly available to support the identification of possible potential
15143 vulnerabilities.

15144 The remaining implicit evaluation evidence for this sub-activity depends on the components that
15145 have been included in the assurance package. The evidence provided for each component is to be
15146 used as input in this sub-activity.

15147 Other input for this sub-activity is:

- 15148 a) current information regarding public domain potential vulnerabilities and attacks (e.g.
15149 from an evaluation authority).

15150 **16.2.2.3 Application notes**

15151 The evaluator should consider performing additional tests as a result of potential vulnerabilities
15152 encountered during other parts of the evaluation.

15153 **16.2.2.4 Action AVA_VAN.2.1E**

15154 ISO/IEC 15408-3 AVA_VAN.2.1C: *The TOE shall be suitable for testing.*

15155 **16.2.2.4.1 Work unit AVA_VAN.2-1**

15156 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
15157 the configuration under evaluation as specified in the ST.

15158 The TOE provided by the developer and identified in the test plan should have the same unique
15159 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
15160 introduction.

15161 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
15162 comprise a number of distinct hardware and software entities that need to be tested in accordance
15163 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

15164 The evaluator should consider the security objectives for the operational environment described in
15165 the ST that may apply to the test environment and ensure they are met in the testing environment.
15166 There may be some objectives for the operational environment that do not apply to the test
15167 environment. For example, an objective about user clearances may not apply; however, an
15168 objective about a single point of connection to a network would apply.

15169 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
15170 ensure that these resources are calibrated correctly.

15171 **16.2.2.4.2 Work unit AVA_VAN.2-2**

15172 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a
15173 known state

15174 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
15175 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
15176 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
15177 installed properly and is in a known state. If this is not the case, then the evaluator should follow
15178 the developer's procedures to install and start up the TOE, using the supplied guidance only.

15179 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
15180 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.

15181 **16.2.2.5 Action AVA_VAN.2.2E**

15182 **16.2.2.5.1 Work unit AVA_VAN.2-3**

15183 The evaluator ***shall examine*** sources of information publicly available to identify potential
15184 vulnerabilities in the TOE.

15185 The evaluator examines the sources of information publicly available to support the identification
15186 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
15187 information which the evaluator should consider using items such as those available on the world
15188 wide web, including:

15189 a) specialist publications (magazines, books);

15190 b) research papers.

15191 The evaluator should not constrain their consideration of publicly available information to the
15192 above, but should consider any other relevant information available.

15193 While examining the evidence provided the evaluator will use the information in the public domain
15194 to further search for potential vulnerabilities. Where the evaluators have identified areas of

- 15195 concern, the evaluator should consider information publicly available that relate to those areas of
15196 concern.
- 15197 The availability of information that may be readily available to an attacker that helps to identify
15198 and facilitate attacks may substantially enhance the attack potential of a given attacker. The
15199 accessibility of vulnerability information and sophisticated attack tools on the Internet makes it
15200 more likely that this information will be used in attempts to identify potential vulnerabilities in the
15201 TOE and exploit them. Modern search tools make such information easily available to the evaluator,
15202 and the determination of resistance to published potential vulnerabilities and well known generic
15203 attacks can be achieved in a cost-effective manner.
- 15204 The search of the information publicly available should be focused on those sources that refer
15205 specifically to the product from which the TOE is derived. The extensiveness of this search should
15206 consider the following factors: TOE type, evaluator experience in this TOE type, expected attack
15207 potential and the level of ADV evidence available.
- 15208 The identification process is iterative, where the identification of one potential vulnerability may
15209 lead to identifying another area of concern that requires further investigation.
- 15210 The evaluator will report what actions were taken to identify potential vulnerabilities in the
15211 evidence. However, in this type of search, the evaluator may not be able to describe the steps in
15212 identifying potential vulnerabilities before the outset of the examination, as the approach may
15213 evolve as a result of findings during the search.
- 15214 The evaluator will report the evidence examined in completing the search for potential
15215 vulnerabilities. This selection of evidence may be derived from those areas of concern identified by
15216 the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to
15217 another rationale provided by the evaluator.
- 15218 **16.2.2.6 Action AVA_VAN.2.3E**
- 15219 **16.2.2.6.1 Work unit AVA_VAN.2-4**
- 15220 The evaluator ***shall conduct*** a search of ST, guidance documentation, functional specification, TOE
15221 design and security architecture description evidence to identify possible potential vulnerabilities
15222 in the TOE.
- 15223 A search of the evidence should be completed whereby specifications and documentation for the
15224 TOE are analysed and then potential vulnerabilities in the TOE are hypothesised, or speculated.
15225 The list of hypothesised potential vulnerabilities is then prioritised on the basis of the estimated
15226 probability that a potential vulnerability exists and, assuming an exploitable vulnerability does
15227 exist the attack potential required to exploit it, and on the extent of control or compromise it would
15228 provide. The prioritised list of potential vulnerabilities is used to direct penetration testing against
15229 the TOE.
- 15230 The security architecture description provides the developer vulnerability analysis, as it
15231 documents how the TSF protects itself from interference from untrusted subjects and prevents the
15232 bypass of security enforcement functionality. Therefore, the evaluator should use this description
15233 of the protection of the TSF as a basis for the search for possible ways to undermine the TSF.
- 15234 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent
15235 vulnerability analysis should consider generic potential vulnerabilities under each of the following
15236 headings:
- 15237 a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be
15238 supplied by the evaluation authority;
- 15239 b) bypassing;

- 15240 c) tampering;
- 15241 d) direct attacks;
- 15242 e) monitoring;
- 15243 f) misuse.
- 15244 Items b) - f) are explained in greater detail in Annex B.
- 15245 The security architecture description should be considered in light of each of the above generic
15246 potential vulnerabilities. Each potential vulnerability should be considered to search for possible
15247 ways in which to defeat the TSF protection and undermine the TSF.
- 15248 **16.2.2.6.2 Work unit AVA_VAN.2-5**
- 15249 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates
15250 for testing and applicable to the TOE in its operational environment.
- 15251 It may be identified that no further consideration of the potential vulnerability is required if for
15252 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
15253 prevent exploitation of the potential vulnerability in that operational environment. For instance,
15254 restricting physical access to the TOE to authorised users only may effectively render a potential
15255 vulnerability to tampering unexploitable.
- 15256 The evaluator records any reasons for exclusion of potential vulnerabilities from further
15257 consideration if the evaluator determines that the potential vulnerability is not applicable in the
15258 operational environment. Otherwise the evaluator records the potential vulnerability for further
15259 consideration.
- 15260 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
15261 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.
- 15262 **16.2.2.7 Action AVA_VAN.2.4E**
- 15263 **16.2.2.7.1 Work unit AVA_VAN.2-6**
- 15264 The evaluator ***shall devise*** penetration tests, based on the independent search for potential
15265 vulnerabilities.
- 15266 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
15267 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
15268 the sources of information publicly available. Any current information provided to the evaluator by
15269 a third party (e.g. evaluation authority) regarding known potential vulnerabilities will be
15270 considered by the evaluator, together with any encountered potential vulnerabilities resulting
15271 from the performance of other evaluation activities.
- 15272 The evaluator is reminded that, as for considering the security architecture description in the
15273 search for vulnerabilities (as detailed in AVA_VAN.2-4), testing should be performed to confirm the
15274 architectural properties. This is likely to require negative tests attempting to disprove the
15275 properties of the security architecture. In developing the strategy for penetration testing, the
15276 evaluator will ensure that each of the major characteristics of the security architecture description
15277 are tested, either in functional testing (as considered in 15) or evaluator penetration testing.
- 15278 The evaluator will probably find it practical to carry out penetration test using a series of test cases,
15279 where each test case will test for a specific potential vulnerability.

- 15280 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15281 domain) beyond those which required a Basic attack potential. In some cases, however, it will be
15282 necessary to carry out a test before the exploitability can be determined. Where, as a result of
15283 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Basic
15284 attack potential, this is reported in the ETR as a residual vulnerability.
- 15285 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be
15286 found in Annex B.4.
- 15287 Potential vulnerabilities hypothesised as exploitable only by attackers possessing Enhanced-Basic,
15288 Moderate or High attack potential do not result in a failure of this evaluator action. Where analysis
15289 supports the hypothesis, these need not be considered further as an input to penetration testing.
15290 However, such vulnerabilities are reported in the ETR as residual vulnerabilities.
- 15291 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Basic attack
15292 potential and resulting in a violation of the security objectives should be the highest priority
15293 potential vulnerabilities comprising the list used to direct penetration testing against the TOE.
- 15294 **16.2.2.7.2 Work unit AVA_VAN.2-7**
- 15295 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of
15296 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
15297 documentation shall include:
- 15298 a) identification of the potential vulnerability the TOE is being tested for;
 - 15299 b) instructions to connect and setup all required test equipment as required to conduct the
15300 penetration test;
 - 15301 c) instructions to establish all penetration test prerequisite initial conditions;
 - 15302 d) instructions to stimulate the TSF;
 - 15303 e) instructions for observing the behaviour of the TSF;
 - 15304 f) descriptions of all expected results and the necessary analysis to be performed on the
15305 observed behaviour for comparison against expected results;
 - 15306 g) instructions to conclude the test and establish the necessary post-test state for the TOE.
- 15307 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
15308 identified during the search of the public domain and the analysis of the evaluation evidence.
- 15309 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
15310 those for which a Basic attack potential is required to effect an attack. However, as a result of
15311 evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only
15312 by an attacker with greater than Basic attack potential. Such vulnerabilities are to be reported in
15313 the ETR as residual vulnerabilities.
- 15314 With an understanding of the potential vulnerability, the evaluator determines the most feasible
15315 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 15316 a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe
15317 responses (It is possible that the evaluator will need to use an interface to the TOE other
15318 than the TSFI to demonstrate properties of the TSF such as those described in the
15319 security architecture description (as required by ADV_ARC). It should be noted, that
15320 although these TOE interfaces provide a means of testing the TSF properties, they are not
15321 the subject of the test.);

- 15322 b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects
15323 that will need to exist and security attributes they will need to have);
- 15324 c) special test equipment that will be required to either stimulate a TSFI or make
15325 observations of a TSFI (although it is unlikely that specialist equipment would be
15326 required to exploit a potential vulnerability assuming a Basic attack potential);
- 15327 d) whether theoretical analysis should replace physical testing, particularly relevant where
15328 the results of an initial test can be extrapolated to demonstrate that repeated attempts of
15329 an attack are likely to succeed after a given number of attempts.
- 15330 The evaluator will probably find it practical to carry out penetration testing using a series of test
15331 cases, where each test case will test for a specific potential vulnerability.
- 15332 The intent of specifying this level of detail in the test documentation is to allow another evaluator
15333 to repeat the tests and obtain an equivalent result.
- 15334 **16.2.2.7.3 Work unit AVA_VAN.2-8**
- 15335 The evaluator ***shall conduct*** penetration testing.
- 15336 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.2-6 as a
15337 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
15338 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
15339 as a result of information learnt during penetration testing that, if performed by the evaluator, are
15340 to be recorded in the penetration test documentation. Such tests may be required to follow up
15341 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
15342 evaluator during the pre-planned testing.
- 15343 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the
15344 evaluator should determine whether or not the evaluator's own analysis was incorrect, or if
15345 evaluation deliverables are incorrect or incomplete.
- 15346 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15347 domain) beyond those which required a Basic attack potential. In some cases, however, it will be
15348 necessary to carry out a test before the exploitability can be determined. Where, as a result of
15349 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond basic
15350 attack potential, this is reported in the ETR as a residual vulnerability.
- 15351 **16.2.2.7.4 Work unit AVA_VAN.2-9**
- 15352 The evaluator ***shall record*** the actual results of the penetration tests.
- 15353 While some specific details of the actual test results may be different from those expected (e.g. time
15354 and date fields in an audit record) the overall result should be identical. Any unexpected test
15355 results should be investigated. The impact on the evaluation should be stated and justified.
- 15356 **16.2.2.7.5 Work unit AVA_VAN.2-10**
- 15357 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing
15358 approach, configuration, depth and results.
- 15359 The penetration testing information reported in the ETR allows the evaluator to convey the overall
15360 penetration testing approach and effort expended on this sub-activity. The intent of providing this
15361 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
15362 intended that the information regarding penetration testing in the ETR be an exact reproduction of
15363 specific test steps or results of individual penetration tests. The intention is to provide enough
15364 detail to allow other evaluators and evaluation authorities to gain some insight about the

15365 penetration testing approach chosen, amount of penetration testing performed, TOE test
15366 configurations, and the overall results of the penetration testing activity.

15367 Information that would typically be found in the ETR subclause regarding evaluator penetration
15368 testing efforts is:

15369 a) TOE test configurations. The particular configurations of the TOE that were penetration
15370 tested;

15371 b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the
15372 focus of the penetration testing;

15373 c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.

15374 This list is by no means exhaustive and is only intended to provide some context as to the type of
15375 information that should be present in the ETR concerning the penetration testing the evaluator
15376 performed during the evaluation.

15377 **16.2.2.7.6 Work unit AVA_VAN.2-11**

15378 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its
15379 operational environment, is resistant to an attacker possessing a Basic attack potential.

15380 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
15381 an attacker possessing less than an Enhanced-Basic attack potential, then this evaluator action fails.

15382 The guidance in B.4 should be used to determine the attack potential required to exploit a
15383 particular vulnerability and whether it can therefore be exploited in the intended environment. It
15384 may not be necessary for the attack potential to be calculated in every instance, only if there is
15385 some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an
15386 attack potential less than Enhanced-Basic.

15387 **16.2.2.7.7 Work unit AVA_VAN.2-12**

15388 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities,
15389 detailing for each:

15390 a) its source (e.g. evaluation methodology activity being undertaken when it was conceived,
15391 known to the evaluator, read in a publication);

15392 b) the SFR(s) not met;

15393 c) a description;

15394 d) whether it is exploitable in its operational environment or not (i.e. exploitable or
15395 residual).

15396 e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity
15397 and the equipment required to perform the identified vulnerabilities, and the
15398 corresponding values using the tables B.2 and B.3 of Annex B.4.

15399 **16.2.3 Evaluation of sub-activity (AVA_VAN.3)**

15400 **16.2.3.1 Objectives**

15401 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
15402 has vulnerabilities exploitable by attackers possessing Enhanced-Basic attack potential.

15403 **16.2.3.2 Input**

15404 The evaluation evidence for this sub-activity is:

15405 a) the ST;

15406 b) the functional specification;

15407 c) the TOE design;

15408 d) the security architecture description;

15409 e) the implementation subset selected;

15410 f) the guidance documentation;

15411 g) the TOE suitable for testing;

15412 h) information publicly available to support the identification of possible potential
15413 vulnerabilities;

15414 i) the results of the testing of the basic design.

15415 The remaining implicit evaluation evidence for this sub-activity depends on the components that
15416 have been included in the assurance package. The evidence provided for each component is to be
15417 used as input in this sub-activity.

15418 Other input for this sub-activity is:

15419 a) current information regarding public domain potential vulnerabilities and attacks (e.g.
15420 from an evaluation authority).

15421 **16.2.3.3 Application notes**

15422 During the conduct of evaluation activities the evaluator may also identify areas of concern. These
15423 are specific portions of the TOE evidence that the evaluator has some reservation about, although
15424 the evidence meets the requirements for the activity with which the evidence is associated. For
15425 example, a particular interface specification looks particularly complex, and therefore may be
15426 prone to error either in the development of the TOE or in the operation of the TOE. There is no
15427 potential vulnerability apparent at this stage, further investigation is required. This is beyond the
15428 bounds of encountered, as further investigation is required.

15429 The focused approach to the identification of potential vulnerabilities is an analysis of the evidence
15430 with the aim of identifying any potential vulnerabilities evident through the contained information.
15431 It is an unstructured analysis, as the approach is not predetermined. Further guidance on focused
15432 vulnerability analysis can be found in Annex B.2.2.2.2.

15433 **16.2.3.4 Action AVA_VAN.3.1E**

15434 ISO/IEC 15408-3 AVA_VAN.3.1C: *The TOE shall be suitable for testing.*

15435 **16.2.3.4.1 Work unit AVA_VAN.3-1**

15436 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
15437 the configuration under evaluation as specified in the ST.

- 15438 The TOE provided by the developer and identified in the test plan should have the same unique
 15439 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
 15440 introduction.
- 15441 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
 15442 comprise a number of distinct hardware and software entities that need to be tested in accordance
 15443 with the ST. The evaluator verifies that all test configurations are consistent with the ST.
- 15444 The evaluator should consider the security objectives for the operational environment described in
 15445 the ST that may apply to the test environment and ensure they are met in the testing environment.
 15446 There may be some objectives for the operational environment that do not apply to the test
 15447 environment. For example, an objective about user clearances may not apply; however, an
 15448 objective about a single point of connection to a network would apply.
- 15449 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
 15450 ensure that these resources are calibrated correctly.
- 15451 **16.2.3.4.2 Work unit AVA_VAN.3-2**
- 15452 The evaluator *shall examine* the TOE to determine that it has been installed properly and is in a
 15453 known state
- 15454 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
 15455 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
 15456 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
 15457 installed properly and is in a known state. If this is not the case, then the evaluator should follow
 15458 the developer's procedures to install and start up the TOE, using the supplied guidance only.
- 15459 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
 15460 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.
- 15461 **16.2.3.5 Action AVA_VAN.3.2E**
- 15462 **16.2.3.5.1 Work unit AVA_VAN.3-3**
- 15463 The evaluator *shall examine* sources of information publicly available to identify potential
 15464 vulnerabilities in the TOE.
- 15465 The evaluator examines the sources of information publicly available to support the identification
 15466 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
 15467 information which the evaluator should consider using items such as those available on the world
 15468 wide web, including:
- 15469 a) specialist publications (magazines, books);
 - 15470 b) research papers;
 - 15471 c) conference proceedings.
- 15472 The evaluator should not constrain their consideration of publicly available information to the
 15473 above, but should consider any other relevant information available.
- 15474 While examining the evidence provided the evaluator will use the information in the public domain
 15475 to further search for potential vulnerabilities. Where the evaluators have identified areas of
 15476 concern, the evaluator should consider information publicly available that relate to those areas of
 15477 concern.

15478 The availability of information that may be readily available to an attacker that helps to identify
 15479 and facilitate attacks may substantially enhance the attack potential of a given attacker. The
 15480 accessibility of vulnerability information and sophisticated attack tools on the Internet makes it
 15481 more likely that this information will be used in attempts to identify potential vulnerabilities in the
 15482 TOE and exploit them. Modern search tools make such information easily available to the evaluator,
 15483 and the determination of resistance to published potential vulnerabilities and well known generic
 15484 attacks can be achieved in a cost-effective manner.

15485 The search of the information publicly available should be focused on those sources that refer to
 15486 the technologies used in the development of the product from which the TOE is derived. The
 15487 extensiveness of this search should consider the following factors: TOE type, evaluator experience
 15488 in this TOE type, expected attack potential and the level of ADV evidence available.

15489 The identification process is iterative, where the identification of one potential vulnerability may
 15490 lead to identifying another area of concern that requires further investigation.

15491 The evaluator will report what actions were taken to identify potential vulnerabilities in the
 15492 evidence. However, in this type of search, the evaluator may not be able to describe the steps in
 15493 identifying potential vulnerabilities before the outset of the examination, as the approach may
 15494 evolve as a result of findings during the search.

15495 The evaluator will report the evidence examined in completing the search for potential
 15496 vulnerabilities. This selection of evidence may be derived from those areas of concern identified by
 15497 the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to
 15498 another rationale provided by the evaluator.

15499 **16.2.3.6 Action AVA_VAN.3.3E**

15500 **16.2.3.6.1 Work unit AVA_VAN.3-4**

15501 The evaluator ***shall conduct*** a focused search of ST, guidance documentation, functional
 15502 specification, TOE design, security architecture description and implementation representation to
 15503 identify possible potential vulnerabilities in the TOE.

15504 A flaw hypothesis methodology needs to be used whereby specifications and development and
 15505 guidance evidence are analysed and then potential vulnerabilities in the TOE are hypothesised, or
 15506 speculated.

15507 The evaluator uses the knowledge of the TOE design and operation gained from the TOE
 15508 deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE
 15509 and potential errors in the specified method of operation of the TOE.

15510 The security architecture description provides the developer vulnerability analysis, as it
 15511 documents how the TSF protects itself from interference from untrusted subjects and prevents the
 15512 bypass of security enforcement functionality. Therefore, the evaluator should build upon the
 15513 understanding of the TSF protection gained from the analysis of this evidence and then develop
 15514 this in the knowledge gained from other development ADV evidence.

15515 The approach taken is directed by areas of concern identified during examination of the evidence
 15516 during the conduct of evaluation activities and ensuring a representative sample of the
 15517 development and guidance evidence provided for the evaluation is searched.

15518 For guidance on sampling see Annex A.2. This guidance should be considered when selecting the
 15519 subset, giving reasons for:

15520 a) the approach used in selection;

15521 b) qualification that the evidence to be examined supports that approach.

- 15522 The areas of concern may relate to the sufficiency of specific protection features detailed in the
15523 security architecture description.
- 15524 The evidence to be considered during the vulnerability analysis may be linked to the evidence the
15525 attacker is assumed to be able to obtain. For example, the developer may protect the TOE design
15526 and implementation representations, so the only information assumed to be available to an
15527 attacker is the functional specification and guidance (publicly available). So, although the
15528 objectives for assurance in the TOE ensure the TOE design and implementation representation
15529 requirements are met, these design representations may only be searched to further investigate
15530 areas of concerns.
- 15531 On the other hand, if the source is publicly available it would be reasonable to assume that the
15532 attacker has access to the source and can use this in attempts to attack the TOE. Therefore, the
15533 source should be considered in the focused examination approach.
- 15534 The following indicates examples for the selection of the subset of evidence to be considered:
- 15535 a) For an evaluation where all levels of design abstraction from functional specification to
15536 implementation representation are provided, examination of information in the
15537 functional specification and the implementation representation may be selected, as the
15538 functional specification provides detail of interfaces available to an attacker, and the
15539 implementation representation incorporates the design decisions made at all other
15540 design abstractions. Therefore, the TOE design information will be considered as part of
15541 the implementation representation.
 - 15542 b) Examination of a particular subset of information in each of the design representations
15543 provided for the evaluation.
 - 15544 c) Coverage of particular SFRs through each of the design representations provided for the
15545 evaluation.
 - 15546 d) Examination of each of the design representations provided for the evaluation,
15547 considering different SFRs within each design representations.
 - 15548 e) Examination of aspects of the evidence provided for the evaluation relating to current
15549 potential vulnerability information the evaluator has received (e.g. from a scheme).
- 15550 This approach to identification of potential vulnerabilities is to take an ordered and planned
15551 approach; applying a system to the examination. The evaluator is to describe the method to be used
15552 in terms of what evidence will be considered, the information within the evidence that is to be
15553 examined, the manner in which this information is to be considered and the hypothesis that is to be
15554 created.
- 15555 The following provide some examples that a hypothesis may take:
- 15556 a) consideration of malformed input for interfaces available to an attacker at the external
15557 interfaces;
 - 15558 b) examination of a key security mechanism cited in the security architecture description,
15559 such as process separation, hypothesising internal buffer overflows that may lead to
15560 degradation of separation;
 - 15561 c) search to identify any objects created in the TOE implementation representation that are
15562 then not fully controlled by the TSF, and could be used by an attacker to undermine SFRs.
- 15563 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE
15564 and specify an approach to the search that "all interface specifications provided in the functional

- 15565 specification and TOE design will be searched to hypothesise potential vulnerabilities” and go on to
15566 explain the methods used in the hypothesis.
- 15567 The identification process is iterative, where the identification of one potential vulnerability may
15568 lead to identifying another area of concern that requires further investigation.
- 15569 The evaluator will report what actions were taken to identify potential vulnerabilities in the
15570 evidence. However, in this type of search, the evaluator may not be able to describe the steps in
15571 identifying potential vulnerabilities before the outset of the examination, as the approach may
15572 evolve as a result of findings during the search.
- 15573 The evaluator will report the evidence examine in completing the search for potential
15574 vulnerabilities. This selection of evidence may be derived from those areas of concern identified by
15575 the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to
15576 another rationale provided by the evaluator.
- 15577 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent
15578 vulnerability analysis should consider generic potential vulnerabilities under each of the following
15579 headings:
- 15580 a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be
15581 supplied by the evaluation authority;
- 15582 b) bypassing;
- 15583 c) tampering;
- 15584 d) direct attacks;
- 15585 e) monitoring;
- 15586 f) misuse.
- 15587 Items b) - f) are explained in greater detail in Annex B.
- 15588 The security architecture description should be considered in light of each of the above generic
15589 potential vulnerabilities. Each potential vulnerability should be considered to search for possible
15590 ways in which to defeat the TSF protection and undermine the TSF.
- 15591 **16.2.3.6.2 Work unit AVA_VAN.3-5**
- 15592 The evaluator **shall record** in the ETR the identified potential vulnerabilities that are candidates
15593 for testing and applicable to the TOE in its operational environment.
- 15594 It may be identified that no further consideration of the potential vulnerability is required if for
15595 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
15596 prevent exploitation of the potential vulnerability in that operational environment. For instance,
15597 restricting physical access to the TOE to authorised users only may effectively render a potential
15598 vulnerability to tampering unexploitable.
- 15599 The evaluator records any reasons for exclusion of potential vulnerabilities from further
15600 consideration if the evaluator determines that the potential vulnerability is not applicable in the
15601 operational environment. Otherwise the evaluator records the potential vulnerability for further
15602 consideration.
- 15603 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
15604 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15605 **16.2.3.7 Action AVA_VAN.3.4E**15606 **16.2.3.7.1 Work unit AVA_VAN.3-6**

15607 The evaluator ***shall devise*** penetration tests, based on the independent search for potential
15608 vulnerabilities.

15609 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
15610 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
15611 the sources of information publicly available. Any current information provided to the evaluator by
15612 a third party (e.g. evaluation authority) regarding known potential vulnerabilities will be
15613 considered by the evaluator, together with any encountered potential vulnerabilities resulting
15614 from the performance of other evaluation activities.

15615 The evaluator is reminded that, as for considering the security architecture description in the
15616 search for vulnerabilities (as detailed in AVA_VAN.3-4), testing should be performed to confirm the
15617 architectural properties. If requirements from ATE_DPT are included in the SARs, the developer
15618 testing evidence will include testing performed to confirm the correct implementation of any
15619 specific mechanisms detailed in the security architecture description. However, the developer
15620 testing will not necessarily include testing of all aspects of the architectural properties that protect
15621 the TSF, as much of this testing will be negative testing in nature, attempting to disprove the
15622 properties. In developing the strategy for penetration testing, the evaluator will ensure that all
15623 aspects of the security architecture description are tested, either in functional testing (as
15624 considered in 15) or evaluator penetration testing.

15625 It will probably be practical to carry out penetration test using a series of test cases, where each
15626 test case will test for a specific potential vulnerability.

15627 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15628 domain) beyond those which required an Enhanced-Basic attack potential. In some cases, however,
15629 it will be necessary to carry out a test before the exploitability can be determined. Where, as a
15630 result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond
15631 Enhanced-Basic attack potential, this is reported in the ETR as a residual vulnerability.

15632 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be
15633 found in Annex B.4.

15634 Potential vulnerabilities hypothesised as exploitable only by attackers possessing Moderate or
15635 High attack potential do not result in a failure of this evaluator action. Where analysis supports the
15636 hypothesis, these need not be considered further as an input to penetration testing. However, such
15637 vulnerabilities are reported in the ETR as residual vulnerabilities.

15638 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Basic or
15639 Enhanced-Basic attack potential and resulting in a violation of the security objectives should be the
15640 highest priority potential vulnerabilities comprising the list used to direct penetration testing
15641 against the TOE.

15642 **16.2.3.7.2 Work unit AVA_VAN.3-7**

15643 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of
15644 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
15645 documentation shall include:

- 15646 a) identification of the potential vulnerability the TOE is being tested for;
- 15647 b) instructions to connect and setup all required test equipment as required to conduct the
15648 penetration test;

- 15649 c) instructions to establish all penetration test prerequisite initial conditions;
- 15650 d) instructions to stimulate the TSF;
- 15651 e) instructions for observing the behaviour of the TSF;
- 15652 f) descriptions of all expected results and the necessary analysis to be performed on the
15653 observed behaviour for comparison against expected results;
- 15654 g) instructions to conclude the test and establish the necessary post-test state for the TOE.
- 15655 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
15656 identified during the search of the public domain and the analysis of the evaluation evidence.
- 15657 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
15658 those for which an Enhanced-Basic attack potential is required to effect an attack. However, as a
15659 result of evaluation expertise, the evaluator may discover a potential vulnerability that is
15660 exploitable only by an attacker with greater than Enhanced-Basic attack potential. Such
15661 vulnerabilities are to be reported in the ETR as residual vulnerabilities.
- 15662 With an understanding of the potential vulnerability, the evaluator determines the most feasible
15663 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 15664 a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe
15665 responses (It is possible that the evaluator will need to use an interface to the TOE other
15666 than the TSFI to demonstrate properties of the TSF such as those described in the
15667 security architecture description (as required by ADV_ARC). It should be noted, that
15668 although these TOE interfaces provide a means of testing the TSF properties, they are not
15669 the subject of the test.);
- 15670 b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects
15671 that will need to exist and security attributes they will need to have);
- 15672 c) special test equipment that will be required to either stimulate a TSFI or make
15673 observations of a TSFI (although it is unlikely that specialist equipment would be
15674 required to exploit a potential vulnerability assuming an Enhanced-Basic attack
15675 potential);
- 15676 d) whether theoretical analysis should replace physical testing, particularly relevant where
15677 the results of an initial test can be extrapolated to demonstrate that repeated attempts of
15678 an attack are likely to succeed after a given number of attempts.
- 15679 The evaluator will probably find it practical to carry out penetration testing using a series of test
15680 cases, where each test case will test for a specific potential vulnerability.
- 15681 The intent of specifying this level of detail in the test documentation is to allow another evaluator
15682 to repeat the tests and obtain an equivalent result.
- 15683 **16.2.3.7.3 Work unit AVA_VAN.3-8**
- 15684 The evaluator ***shall conduct*** penetration testing.
- 15685 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.3-6 as a
15686 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
15687 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
15688 as a result of information learnt during penetration testing that, if performed by the evaluator, are
15689 to be recorded in the penetration test documentation. Such tests may be required to follow up

- 15690 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
15691 evaluator during the pre-planned testing.
- 15692 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the
15693 evaluator should determine whether or not the evaluator's own analysis was incorrect, or if
15694 evaluation deliverables are incorrect or incomplete.
- 15695 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15696 domain) beyond those which required an Enhanced-Basic attack potential. In some cases, however,
15697 it will be necessary to carry out a test before the exploitability can be determined. Where, as a
15698 result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond
15699 Enhanced-Basic attack potential, this is reported in the ETR as a residual vulnerability.
- 15700 **16.2.3.7.4 Work unit AVA_VAN.3-9**
- 15701 The evaluator *shall record* the actual results of the penetration tests.
- 15702 While some specific details of the actual test results may be different from those expected (e.g. time
15703 and date fields in an audit record) the overall result should be identical. Any unexpected test
15704 results should be investigated. The impact on the evaluation should be stated and justified.
- 15705 **16.2.3.7.5 Work unit AVA_VAN.3-10**
- 15706 The evaluator *shall report* in the ETR the evaluator penetration testing effort, outlining the testing
15707 approach, configuration, depth and results.
- 15708 The penetration testing information reported in the ETR allows the evaluator to convey the overall
15709 penetration testing approach and effort expended on this sub-activity. The intent of providing this
15710 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
15711 intended that the information regarding penetration testing in the ETR be an exact reproduction of
15712 specific test steps or results of individual penetration tests. The intention is to provide enough
15713 detail to allow other evaluators and evaluation authorities to gain some insight about the
15714 penetration testing approach chosen, amount of penetration testing performed, TOE test
15715 configurations, and the overall results of the penetration testing activity.
- 15716 Information that would typically be found in the ETR subclause regarding evaluator penetration
15717 testing efforts is:
- 15718 a) TOE test configurations. The particular configurations of the TOE that were penetration
15719 tested;
- 15720 b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the
15721 focus of the penetration testing;
- 15722 c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.
- 15723 This list is by no means exhaustive and is only intended to provide some context as to the type of
15724 information that should be present in the ETR concerning the penetration testing the evaluator
15725 performed during the evaluation.
- 15726 **16.2.3.7.6 Work unit AVA_VAN.3-11**
- 15727 The evaluator *shall examine* the results of all penetration testing to determine that the TOE, in its
15728 operational environment, is resistant to an attacker possessing an Enhanced-Basic attack potential.
- 15729 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
15730 an attacker possessing less than Moderate attack potential, then this evaluator action fails.

15731 The guidance in B.4 should be used to determine the attack potential required to exploit a
15732 particular vulnerability and whether it can therefore be exploited in the intended environment. It
15733 may not be necessary for the attack potential to be calculated in every instance, only if there is
15734 some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an
15735 attack potential less than Moderate.

15736 **16.2.3.7.7 Work unit AVA_VAN.3-12**

15737 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities,
15738 detailing for each:

15739 a) its source (e.g. evaluation methodology activity being undertaken when it was conceived,
15740 known to the evaluator, read in a publication);

15741 b) the SFR(s) not met;

15742 c) a description;

15743 d) whether it is exploitable in its operational environment or not (i.e. exploitable or
15744 residual).

15745 e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity
15746 and the equipment required to perform the identified vulnerabilities, and the
15747 corresponding values using the tables B.2 and B.3 of Annex B.4.

15748 **16.2.4 Evaluation of sub-activity (AVA_VAN.4)**

15749 **16.2.4.1 Objectives**

15750 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
15751 has vulnerabilities exploitable by attackers possessing Moderate attack potential.

15752 **16.2.4.2 Input**

15753 The evaluation evidence for this sub-activity is:

15754 f) the ST;

15755 g) the functional specification;

15756 h) the TOE design;

15757 i) the security architecture description;

15758 j) the implementation representation;

15759 k) the guidance documentation;

15760 l) the TOE suitable for testing;

15761 m) information publicly available to support the identification of possible potential
15762 vulnerabilities;

15763 n) the results of the testing of the basic design.

15764 The remaining implicit evaluation evidence for this sub-activity depends on the components that
15765 have been included in the assurance package. The evidence provided for each component is to be
15766 used as input in this sub-activity.

15767 Other input for this sub-activity is:

- 15768 a) current information regarding public domain potential vulnerabilities and attacks (e.g.
15769 from an evaluation authority).

15770 **16.2.4.3 Application notes**

15771 The methodical analysis approach takes the form of a structured examination of the evidence. This
15772 method requires the evaluator to specify the structure and form the analysis will take (i.e. the
15773 manner in which the analysis is performed is predetermined, unlike the focused analysis). The
15774 method is specified in terms of the information that will be considered and how/why it will be
15775 considered. Further guidance on methodical vulnerability analysis can be found in Annex B.2.2.2.3.

15776 **16.2.4.4 Action AVA_VAN.4.1E**

15777 ISO/IEC 15408-3 AVA_VAN.4.1C: *The TOE shall be suitable for testing.*

15778 **16.2.4.4.1 Work unit AVA_VAN.4-1**

15779 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
15780 the configuration under evaluation as specified in the ST.

15781 The TOE provided by the developer and identified in the test plan should have the same unique
15782 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
15783 introduction.

15784 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
15785 comprise a number of distinct hardware and software entities that need to be tested in accordance
15786 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

15787 The evaluator should consider the security objectives for the operational environment described in
15788 the ST that may apply to the test environment and ensure they are met in the testing environment.
15789 There may be some objectives for the operational environment that do not apply to the test
15790 environment. For example, an objective about user clearances may not apply; however, an
15791 objective about a single point of connection to a network would apply.

15792 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
15793 ensure that these resources are calibrated correctly.

15794 **16.2.4.4.2 Work unit AVA_VAN.4-2**

15795 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a
15796 known state

15797 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
15798 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
15799 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
15800 installed properly and is in a known state. If this is not the case, then the evaluator should follow
15801 the developer's procedures to install and start up the TOE, using the supplied guidance only.

15802 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
15803 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.

15804 **16.2.4.5 Action AVA_VAN.4.2E**

15805 **16.2.4.5.1 Work unit AVA_VAN.4-3**

15806 The evaluator ***shall examine*** sources of information publicly available to identify potential
15807 vulnerabilities in the TOE.

15808 The evaluator examines the sources of information publicly available to support the identification
15809 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
15810 information which the evaluator should consider using items such as those available on the world
15811 wide web, including:

15812 b) specialist publications (magazines, books);

15813 c) research papers;

15814 d) conference proceedings.

15815 The evaluator should not constrain their consideration of publicly available information to the
15816 above, but should consider any other relevant information available.

15817 While examining the evidence provided the evaluator will use the information in the public domain
15818 to further search for potential vulnerabilities. Where the evaluators have identified areas of
15819 concern, the evaluator should consider information publicly available that relate to those areas of
15820 concern.

15821 The availability of information that may be readily available to an attacker that helps to identify
15822 and facilitate attacks may substantially enhance the attack potential of a given attacker. The
15823 accessibility of vulnerability information and sophisticated attack tools on the Internet makes it
15824 more likely that this information will be used in attempts to identify potential vulnerabilities in the
15825 TOE and exploit them. Modern search tools make such information easily available to the evaluator,
15826 and the determination of resistance to published potential vulnerabilities and well known generic
15827 attacks can be achieved in a cost-effective manner.

15828 The search of the information publicly available should be focused on those sources that refer to
15829 the technologies used in the development of the product from which the TOE is derived. The
15830 extensiveness of this search should consider the following factors: TOE type, evaluator experience
15831 in this TOE type, expected attack potential and the level of ADV evidence available.

15832 The identification process is iterative, where the identification of one potential vulnerability may
15833 lead to identifying another area of concern that requires further investigation.

15834 The evaluator will describe the approach to be taken to identify potential vulnerabilities in the
15835 publicly available material, detailing the search to be performed. This may be driven by factors
15836 such as areas of concern identified by the evaluator, linked to the evidence the attacker is assumed
15837 to be able to obtain. However, it is recognised that in this type of search the approach may further
15838 evolve as a result of findings during the search. Therefore, the evaluator will also report any
15839 actions taken in addition to those described in the approach to further investigate issues thought to
15840 lead to potential vulnerabilities, and will report the evidence examined in completing the search
15841 for potential vulnerabilities.

15842 **16.2.4.6 Action AVA_VAN.4.3E**

15843 **16.2.4.6.1 Work unit AVA_VAN.4-4**

15844 The evaluator ***shall conduct*** a methodical analysis of ST, guidance documentation, functional
15845 specification, TOE design, security architecture description and implementation representation to
15846 identify possible potential vulnerabilities in the TOE.

- 15847 Guidance on methodical vulnerability analysis is provided in Annex B.2.2.2.3.
- 15848 This approach to identification of potential vulnerabilities is to take an ordered and planned
15849 approach. A system is to be applied in the examination. The evaluator is to describe the method to
15850 be used in terms of the manner in which this information is to be considered and the hypothesis
15851 that is to be created.
- 15852 A flaw hypothesis methodology needs to be used whereby the ST, development (functional
15853 specification, TOE design and implementation representation) and guidance evidence are analysed
15854 and then vulnerabilities in the TOE are hypothesised, or speculated.
- 15855 The evaluator uses the knowledge of the TOE design and operation gained from the TOE
15856 deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE
15857 and potential errors in the specified method of operation of the TOE.
- 15858 The security architecture description provides the developer vulnerability analysis, as it
15859 documents how the TSF protects itself from interference from untrusted subjects and prevents the
15860 bypass of security enforcement functionality. Therefore, the evaluator should build upon the
15861 understanding of the TSF protection gained from the analysis of this evidence and then develop
15862 this in the knowledge gained from other development ADV evidence.
- 15863 The approach taken to the methodical search for vulnerabilities is to consider any areas of concern
15864 identified in the results of the evaluator's assessment of the development and guidance evidence.
15865 However, the evaluator should also consider each aspect of the security architecture analysis to
15866 search for any ways in which the protection of the TSF can be undermined. It may be helpful to
15867 structure the methodical analysis on the basis of the material presented in the security architecture
15868 description, introducing concerns from other ADV evidence as appropriate. The analysis can then
15869 be further developed to ensure all other material from the ADV evidence is considered.
- 15870 The following provide some examples of hypotheses that may be created when examining the
15871 evidence:
- 15872 a) consideration of malformed input for interfaces available to an attacker at the external
15873 interfaces;
- 15874 b) examination of a key security mechanism cited in the security architecture description,
15875 such as process separation, hypothesising internal buffer overflows that may lead to
15876 degradation of separation;
- 15877 c) search to identify any objects created in the TOE implementation representation that are
15878 then not fully controlled by the TSF, and could be used by an attacker to undermine SFRs.
- 15879 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE
15880 and specify an approach to the search that 'all interface specifications in the evidence provided will
15881 be searched to hypothesise potential vulnerabilities' and go on to explain the methods used in the
15882 hypothesis.
- 15883 In addition, areas of concern the evaluator has identified during examination of the evidence
15884 during the conduct of evaluation activities. Areas of concern may also be identified during the
15885 conduct of other work units associated with this component, in particular AVA_VAN.4-7,
15886 AVA_VAN.4-5 and AVA_VAN.4-6 where the development and conduct of penetration tests may
15887 identify further areas of concerns for investigation, or potential vulnerabilities.
- 15888 However, examination of only a subset of the development and guidance evidence or their contents
15889 is not permitted in this level of rigour. The approach description should provide a demonstration
15890 that the methodical approach used is complete, providing confidence that the approach used to
15891 search the deliverables has considered all of the information provided in those deliverables.

15892 This approach to identification of potential vulnerabilities is to take an ordered and planned
15893 approach; applying a system to the examination. The evaluator is to describe the method to be used
15894 in terms of how the evidence will be considered; the manner in which this information is to be
15895 considered and the hypothesis that is to be created. This approach should be agreed with the
15896 evaluation authority, and the evaluation authority may provide detail of any additional approaches
15897 the evaluator should take to the vulnerability analysis and identify any additional information that
15898 should be considered by the evaluator.

15899 Although a system to identifying potential vulnerabilities is predefined, the identification process
15900 may still be iterative, where the identification of one potential vulnerability may lead to identifying
15901 another area of concern that requires further investigation.

15902 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent
15903 vulnerability analysis should consider generic potential vulnerabilities under each of the following
15904 headings:

15905 a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be
15906 supplied by the evaluation authority;

15907 b) bypassing;

15908 c) tampering;

15909 d) direct attacks;

15910 e) monitoring;

15911 f) misuse.

15912 Items b) - f) are explained in greater detail in Annex B.

15913 The security architecture description should be considered in light of each of the above generic
15914 potential vulnerabilities. Each potential vulnerability should be considered to search for possible
15915 ways in which to defeat the TSF protection and undermine the TSF.

15916 **16.2.4.6.2 Work unit AVA_VAN.4-5**

15917 The evaluator **shall record** in the ETR the identified potential vulnerabilities that are candidates
15918 for testing and applicable to the TOE in its operational environment.

15919 It may be identified that no further consideration of the potential vulnerability is required if for
15920 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
15921 prevent exploitation of the potential vulnerability in that operational environment. For instance,
15922 restricting physical access to the TOE to authorised users only may effectively render a potential
15923 vulnerability to tampering unexploitable.

15924 The evaluator records any reasons for exclusion of potential vulnerabilities from further
15925 consideration if the evaluator determines that the potential vulnerability is not applicable in the
15926 operational environment. Otherwise the evaluator records the potential vulnerability for further
15927 consideration.

15928 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
15929 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15930 **16.2.4.7 Action AVA_VAN.4.4E**15931 **16.2.4.7.1 Work unit AVA_VAN.4-6**

15932 The evaluator ***shall devise*** penetration tests, based on the independent search for potential
15933 vulnerabilities.

15934 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
15935 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
15936 the sources of information publicly available. Any current information provided to the evaluator by
15937 a third party (e.g. evaluation authority) regarding known potential vulnerabilities will be
15938 considered by the evaluator, together with any encountered potential vulnerabilities resulting
15939 from the performance of other evaluation activities.

15940 The evaluator is reminded that, as for considering the security architecture description in the
15941 search for vulnerabilities (as detailed in AVA_VAN.4-3), testing should be performed to confirm the
15942 architectural properties. If requirements from ATE_DPT are included in the SARs, the developer
15943 testing evidence will include testing performed to confirm the correct implementation of any
15944 specific mechanisms detailed in the security architecture description. However, the developer
15945 testing will not necessarily include testing of all aspects of the architectural properties that protect
15946 the TSF, as much of this testing will be negative testing in nature, attempting to disprove the
15947 properties. In developing the strategy for penetration testing, the evaluator will ensure that all
15948 aspects of the security architecture description are tested, either in functional testing (as
15949 considered in 15) or evaluator penetration testing.

15950 The evaluator will probably find it practical to carry out penetration test using a series of test cases,
15951 where each test case will test for a specific potential vulnerability.

15952 The evaluator is not expected to test for potential vulnerabilities (including those in the public
15953 domain) beyond those which required a Moderate attack potential. In some cases, however, it will
15954 be necessary to carry out a test before the exploitability can be determined. Where, as a result of
15955 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Moderate
15956 attack potential, this is reported in the ETR as a residual vulnerability.

15957 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be
15958 found in Annex B.4.

15959 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Moderate (or less)
15960 attack potential and resulting in a violation of the security objectives should be the highest priority
15961 potential vulnerabilities comprising the list used to direct penetration testing against the TOE.

15962 **16.2.4.7.2 Work unit AVA_VAN.4-7**

15963 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of
15964 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
15965 documentation shall include:

- 15966 g) identification of the potential vulnerability the TOE is being tested for;
- 15967 h) instructions to connect and setup all required test equipment as required to conduct the
15968 penetration test;
- 15969 i) instructions to establish all penetration test prerequisite initial conditions;
- 15970 j) instructions to stimulate the TSF;
- 15971 k) instructions for observing the behaviour of the TSF;

- 15972 l) descriptions of all expected results and the necessary analysis to be performed on the
15973 observed behaviour for comparison against expected results;
- 15974 m) instructions to conclude the test and establish the necessary post-test state for the TOE.
- 15975 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
15976 identified during the search of the public domain and the analysis of the evaluation evidence.
- 15977 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
15978 those for which a Moderate attack potential is required to effect an attack. However, as a result of
15979 evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only
15980 by an attacker with greater than Moderate attack potential. Such vulnerabilities are to be reported
15981 in the ETR as residual vulnerabilities.
- 15982 With an understanding of the potential vulnerability, the evaluator determines the most feasible
15983 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 15984 a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe
15985 responses (It is possible that the evaluator will need to use an interface to the TOE other
15986 than the TSFI to demonstrate properties of the TSF such as those described in the
15987 security architecture description (as required by ADV_ARC). It should be noted, that
15988 although these TOE interfaces provide a means of testing the TSF properties, they are not
15989 the subject of the test.);
- 15990 b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects
15991 that will need to exist and security attributes they will need to have);
- 15992 c) special test equipment that will be required to either stimulate a TSFI or make
15993 observations of a TSFI;
- 15994 d) whether theoretical analysis should replace physical testing, particularly relevant where
15995 the results of an initial test can be extrapolated to demonstrate that repeated attempts of
15996 an attack are likely to succeed after a given number of attempts.
- 15997 The evaluator will probably find it practical to carry out penetration testing using a series of test
15998 cases, where each test case will test for a specific potential vulnerability.
- 15999 The intent of specifying this level of detail in the test documentation is to allow another evaluator
16000 to repeat the tests and obtain an equivalent result.
- 16001 **16.2.4.7.3 Work unit AVA_VAN.4-8**
- 16002 The evaluator ***shall conduct*** penetration testing.
- 16003 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.4-6 as a
16004 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
16005 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
16006 as a result of information learnt during penetration testing that, if performed by the evaluator, are
16007 to be recorded in the penetration test documentation. Such tests may be required to follow up
16008 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
16009 evaluator during the pre-planned testing.
- 16010 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the
16011 evaluator should determine whether or not the evaluator's own analysis was incorrect, or if
16012 evaluation deliverables are incorrect or incomplete.
- 16013 The evaluator is not expected to test for potential vulnerabilities (including those in the public
16014 domain) beyond those which required a Moderate attack potential. In some cases, however, it will

16015 be necessary to carry out a test before the exploitability can be determined. Where, as a result of
 16016 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Moderate
 16017 attack potential, this is reported in the ETR as a residual vulnerability.

16018 **16.2.4.7.4 Work unit AVA_VAN.4-9**

16019 The evaluator ***shall record*** the actual results of the penetration tests.

16020 While some specific details of the actual test results may be different from those expected (e.g. time
 16021 and date fields in an audit record) the overall result should be identical. Any unexpected test
 16022 results should be investigated. The impact on the evaluation should be stated and justified.

16023 **16.2.4.7.5 Work unit AVA_VAN.4-10**

16024 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing
 16025 approach, configuration, depth and results.

16026 The penetration testing information reported in the ETR allows the evaluator to convey the overall
 16027 penetration testing approach and effort expended on this sub-activity. The intent of providing this
 16028 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
 16029 intended that the information regarding penetration testing in the ETR be an exact reproduction of
 16030 specific test steps or results of individual penetration tests. The intention is to provide enough
 16031 detail to allow other evaluators and evaluation authorities to gain some insight about the
 16032 penetration testing approach chosen, amount of penetration testing performed, TOE test
 16033 configurations, and the overall results of the penetration testing activity.

16034 Information that would typically be found in the ETR subclause regarding evaluator penetration
 16035 testing efforts is:

16036 a) TOE test configurations. The particular configurations of the TOE that were penetration
 16037 tested;

16038 b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the
 16039 focus of the penetration testing;

16040 c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.

16041 This list is by no means exhaustive and is only intended to provide some context as to the type of
 16042 information that should be present in the ETR concerning the penetration testing the evaluator
 16043 performed during the evaluation.

16044 **16.2.4.7.6 Work unit AVA_VAN.4-11**

16045 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its
 16046 operational environment, is resistant to an attacker possessing a Moderate attack potential.

16047 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
 16048 an attacker possessing less than a High attack potential, then this evaluator action fails.

16049 The guidance in B.4 should be used to determine the attack potential required to exploit a
 16050 particular vulnerability and whether it can therefore be exploited in the intended environment. It
 16051 may not be necessary for the attack potential to be calculated in every instance, only if there is
 16052 some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an
 16053 attack potential less than High.

16054 **16.2.4.7.7 Work unit AVA_VAN.4-12**

16055 The evaluator **shall report** in the ETR all exploitable vulnerabilities and residual vulnerabilities,
16056 detailing for each:

16057 a) its source (e.g. evaluation methodology activity being undertaken when it was conceived,
16058 known to the evaluator, read in a publication);

16059 b) the SFR(s) not met;

16060 c) a description;

16061 d) whether it is exploitable in its operational environment or not (i.e. exploitable or
16062 residual).

16063 e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity
16064 and the equipment required to perform the identified vulnerabilities, and the
16065 corresponding values using the tables B.2 and B.3 of Annex B.4.

16066 **16.2.5 Evaluation of sub-activity (AVA_VAN.5)**

16067 The work units for the evaluation of the sub-activity AVA_VAN.5 are copied from the work units of
16068 AVA_VAN.4 as far as possible except that the TOE is attacked by attackers possessing High attack
16069 potential.

16070 **16.2.5.1 Objectives**

16071 The objective of this sub-activity is to determine whether the TOE, in its operational environment,
16072 has vulnerabilities exploitable by attackers possessing **High** attack potential.

16073 **16.2.5.2 Input**

16074 The evaluation evidence for this sub-activity is:

16075 f) the ST;

16076 g) the functional specification;

16077 h) the TOE design;

16078 i) the security architecture description;

16079 j) the implementation representation;

16080 k) the guidance documentation;

16081 l) the TOE suitable for testing;

16082 m) information publicly available to support the identification of possible potential
16083 vulnerabilities;

16084 n) the results of the testing of the basic design.

16085 The remaining implicit evaluation evidence for this sub-activity depends on the components that
16086 have been included in the assurance package. The evidence provided for each component is to be
16087 used as input in this sub-activity.

16088 Other input for this sub-activity is:

16089 o) current information regarding public domain potential vulnerabilities and attacks (e.g.
16090 from an evaluation authority).

16091 **16.2.5.3 Application notes**

16092 The methodical analysis approach takes the form of a structured examination of the evidence. This
16093 method requires the evaluator to specify the structure and form the analysis will take (i.e. the
16094 manner in which the analysis is performed is predetermined, unlike the focused analysis). The
16095 method is specified in terms of the information that will be considered and how/why it will be
16096 considered. Further guidance on methodical vulnerability analysis can be found in Annex B.2.2.2.3.

16097 **16.2.5.4 Action AVA_VAN.5.1E**

16098 **AVA_VAN.5.1C**

16099 The TOE shall be suitable for testing.

16100 **16.2.5.4.1 Work unit AVA_VAN.5-1**

16101 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with
16102 the configuration under evaluation as specified in the ST.

16103 The TOE provided by the developer and identified in the test plan should have the same unique
16104 reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST
16105 introduction.

16106 It is possible for the ST to specify more than one configuration for evaluation. The TOE may
16107 comprise a number of distinct hardware and software entities that need to be tested in accordance
16108 with the ST. The evaluator verifies that all test configurations are consistent with the ST.

16109 The evaluator should consider the security objectives for the operational environment described in
16110 the ST that may apply to the test environment and ensure they are met in the testing environment.
16111 There may be some objectives for the operational environment that do not apply to the test
16112 environment. For example, an objective about user clearances may not apply; however, an
16113 objective about a single point of connection to a network would apply.

16114 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to
16115 ensure that these resources are calibrated correctly.

16116 **16.2.5.4.2 Work unit AVA_VAN.5-2**

16117 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a
16118 known state

16119 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example,
16120 previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will
16121 satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was
16122 installed properly and is in a known state. If this is not the case, then the evaluator should follow
16123 the developer's procedures to install and start up the TOE, using the supplied guidance only.

16124 If the evaluator has to perform the installation procedures because the TOE is in an unknown state,
16125 this work unit when successfully completed could satisfy work unit AGD_PRE.1-3.

16126 **16.2.5.5 Action AVA_VAN.5.2E**

16127 **16.2.5.5.1 Work unit AVA_VAN.5-3**

16128 The evaluator ***shall examine*** sources of information publicly available to identify potential
16129 vulnerabilities in the TOE.

16130 The evaluator examines the sources of information publicly available to support the identification
16131 of possible potential vulnerabilities in the TOE. There are many sources of publicly available
16132 information which the evaluator should consider using items such as those available on the world
16133 wide web, including:

16134 p) specialist publications (magazines, books);

16135 q) research papers;

16136 r) conference proceedings.

16137 The evaluator should not constrain their consideration of publicly available information to the
16138 above, but should consider any other relevant information available.

16139 While examining the evidence provided the evaluator will use the information in the public domain
16140 to further search for potential vulnerabilities. Where the evaluators have identified areas of
16141 concern, the evaluator should consider information publicly available that relate to those areas of
16142 concern.

16143 The availability of information that may be readily available to an attacker that helps to identify
16144 and facilitate attacks may substantially enhance the attack potential of a given attacker. The
16145 accessibility of vulnerability information and sophisticated attack tools on the Internet makes it
16146 more likely that this information will be used in attempts to identify potential vulnerabilities in the
16147 TOE and exploit them. Modern search tools make such information easily available to the evaluator,
16148 and the determination of resistance to published potential vulnerabilities and well known generic
16149 attacks can be achieved in a cost-effective manner.

16150 The search of the information publicly available should be focused on those sources that refer to
16151 the technologies used in the development of the product from which the TOE is derived. The
16152 extensiveness of this search should consider the following factors: TOE type, evaluator experience
16153 in this TOE type, expected attack potential and the level of ADV evidence available.

16154 The identification process is iterative, where the identification of one potential vulnerability may
16155 lead to identifying another area of concern that requires further investigation.

16156 The evaluator will describe the approach to be taken to identify potential vulnerabilities in the
16157 publicly available material, detailing the search to be performed. This may be driven by factors
16158 such as areas of concern identified by the evaluator, linked to the evidence the attacker is assumed
16159 to be able to obtain. However, it is recognised that in this type of search the approach may further
16160 evolve as a result of findings during the search. Therefore, the evaluator will also report any
16161 actions taken in addition to those described in the approach to further investigate issues thought to
16162 lead to potential vulnerabilities, and will report the evidence examined in completing the search
16163 for potential vulnerabilities.

16164 **16.2.5.6 Action AVA_VAN.5.3E**

16165 **16.2.5.6.1 Work unit AVA_VAN.5-4**

16166 The evaluator ***shall conduct*** a methodical analysis of ST, guidance documentation, functional
16167 specification, TOE design, security architecture description and implementation representation to
16168 identify possible potential vulnerabilities in the TOE.

- 16169 Guidance on methodical vulnerability analysis is provided in Annex B.2.2.2.3.
- 16170 This approach to identification of potential vulnerabilities is to take an ordered and planned
16171 approach. A system is to be applied in the examination. The evaluator is to describe the method to
16172 be used in terms of the manner in which this information is to be considered and the hypothesis
16173 that is to be created.
- 16174 A flaw hypothesis methodology should be used whereby the ST, development (functional
16175 specification, TOE design and implementation representation) and guidance evidence are analysed
16176 and then vulnerabilities in the TOE are hypothesised, or speculated.
- 16177 The evaluator should use the knowledge of the TOE design and operation gained from the TOE
16178 deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE
16179 and potential errors in the specified method of operation of the TOE.
- 16180 The security architecture description provides the developer vulnerability analysis, as it
16181 documents how the TSF protects itself from interference from untrusted subjects and prevents the
16182 bypass of security enforcement functionality. Therefore, the evaluator should build upon the
16183 understanding of the TSF protection gained from the analysis of this evidence and then develop
16184 this in the knowledge gained from other development (e.g. ADV) evidence.
- 16185 The approach taken to the methodical search for vulnerabilities is to consider any areas of concern
16186 identified in the results of the evaluator's assessment of the development and guidance evidence.
16187 However, the evaluator should also consider each aspect of the security architecture analysis to
16188 search for any ways in which the protection of the TSF can be undermined. It may be helpful to
16189 structure the methodical analysis on the basis of the material presented in the security architecture
16190 description, introducing concerns from other ADV evidence as appropriate. The analysis can then
16191 be further developed to ensure all other material from the ADV evidence is considered.
- 16192 The following provide some examples of hypotheses that may be created when examining the
16193 evidence:
- 16194 consideration of malformed input for interfaces available to an attacker at the external interfaces;
- 16195 examination of a key security mechanism cited in the security architecture description, such as
16196 process separation, hypothesising internal buffer overflows that may lead to degradation of
16197 separation;
- 16198 search to identify any objects created in the TOE implementation representation that are then not
16199 fully controlled by the TSF, and could be used by an attacker to undermine SFRs.
- 16200 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE
16201 and specify an approach to the search that 'all interface specifications in the evidence provided will
16202 be searched to hypothesise potential vulnerabilities' and go on to explain the methods used in the
16203 hypothesis.
- 16204 In addition, areas of concern the evaluator has identified during examination of the evidence
16205 during the conduct of evaluation activities. Areas of concern may also be identified during the
16206 conduct of other work units associated with this component, in particular AVA_VAN.5-7,
16207 AVA_VAN.5-5 and AVA_VAN.5-6) where the development and conduct of penetration tests may
16208 identify further areas of concerns for investigation, or potential vulnerabilities.
- 16209 However, examination of only a subset of the development and guidance evidence or their contents
16210 is not permitted in this level of rigour. The approach description should provide a demonstration
16211 that the methodical approach used is complete, providing confidence that the approach used to
16212 search the deliverables has considered all of the information provided in those deliverables.

16213 This approach to identification of potential vulnerabilities is to take an ordered and planned
16214 approach; applying a system to the examination. The evaluator is to describe the method to be used
16215 in terms of how the evidence will be considered; the manner in which this information is to be
16216 considered and the hypothesis that is to be created. This approach should be agreed with the
16217 evaluation authority, and the evaluation authority should provide detail of any additional
16218 approaches the evaluator should take to the vulnerability analysis and identify any additional
16219 information that should be considered by the evaluator.

16220 Although a system to identifying potential vulnerabilities is predefined, the identification process
16221 may still be iterative, where the identification of one potential vulnerability may lead to identifying
16222 another area of concern that requires further investigation.

16223 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent
16224 vulnerability analysis should consider generic potential vulnerabilities under each of the following
16225 headings:

16226 g) generic potential vulnerabilities relevant for the type of TOE being evaluated, as
16227 may be supplied by the evaluation authority;

16228 h) bypassing;

16229 i) tampering;

16230 j) direct attacks;

16231 k) monitoring;

16232 l) misuse.

16233 Items b) - f) are explained in greater detail in Annex B.2.1.

16234 The security architecture description should be considered in light of each of the above generic
16235 potential vulnerabilities. Each potential vulnerability should be considered to search for possible
16236 ways in which to defeat the TSF protection and undermine the TSF.

16237 **16.2.5.6.2 Work unit AVA_VAN.5-5**

16238 The evaluator **shall record** in the ETR the identified potential vulnerabilities that are candidates
16239 for testing and applicable to the TOE in its operational environment.

16240 It may be identified that no further consideration of the potential vulnerability is required if for
16241 example the evaluator identifies that measures in the operational environment, either IT or non-IT,
16242 prevent exploitation of the potential vulnerability in that operational environment. For instance,
16243 restricting physical access to the TOE to authorised users only may effectively render a potential
16244 vulnerability to tampering unexploitable.

16245 The evaluator records any reasons for exclusion of potential vulnerabilities from further
16246 consideration if the evaluator determines that the potential vulnerability is not applicable in the
16247 operational environment. Otherwise the evaluator records the potential vulnerability for further
16248 consideration.

16249 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be
16250 used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

16251 **16.2.5.7 Action AVA_VAN.5.4E**16252 **16.2.5.7.1 Work unit AVA_VAN.5-6**

16253 The evaluator **shall devise** penetration tests, based on the independent search for potential
16254 vulnerabilities.

16255 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the
16256 TOE, in its operational environment, to the potential vulnerabilities identified during the search of
16257 the sources of publicly available information and the analysis of the TOE guidance and design
16258 evidence. The evaluator should have access to current information (e.g. from the evaluation
16259 authority) regarding known potential vulnerabilities that may not have been considered by the
16260 evaluator.

16261 The evaluator is reminded that, as for considering the security architecture description in the
16262 search for vulnerabilities (as detailed in AVA_VAN.5-3), testing should be performed to confirm the
16263 architectural properties. If requirements from ATE_DPT are included in the SARs, the developer
16264 testing evidence will include testing performed to confirm the correct implementation of any
16265 specific mechanisms detailed in the security architecture description. However, the developer
16266 testing will not necessarily include testing of all aspects of the architectural properties that protect
16267 the TSF, as much of this testing will be negative testing in nature, attempting to disprove the
16268 properties. In developing the strategy for penetration testing, the evaluator will ensure that all
16269 aspects of the security architecture description are tested, either in functional testing (as
16270 considered in 15,) or evaluator penetration testing.

16271 The evaluator will probably find it practical to carry out penetration test using a series of test cases,
16272 where each test case will test for a specific potential vulnerability.

16273 The evaluator is not expected to test for potential vulnerabilities (including those in the public
16274 domain) beyond those which required a **High** attack potential. In some cases, however, it will be
16275 necessary to carry out a test before the exploitability can be determined. Where, as a result of
16276 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond **High**
16277 attack potential, this is reported in the ETR as a residual vulnerability.

16278 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be
16279 found in Annex B.4.

16280 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a **High** (or less)
16281 attack potential and resulting in a violation of the security objectives should be the highest priority
16282 potential vulnerabilities comprising the list used to direct penetration testing against the TOE.

16283 **16.2.5.7.2 Work unit AVA_VAN.5-7**

16284 The evaluator **shall produce** penetration test documentation for the tests based on the list of
16285 potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test
16286 documentation shall include:

- 16287 s) identification of the potential vulnerability the TOE is being tested for;
- 16288 t) instructions to connect and setup all required test equipment as required to conduct the
16289 penetration test;
- 16290 u) instructions to establish all penetration test prerequisite initial conditions;
- 16291 v) instructions to stimulate the TSF;
- 16292 w) instructions for observing the behaviour of the TSF;

- 16293 x) descriptions of all expected results and the necessary analysis to be performed on the
16294 observed behaviour for comparison against expected results;
- 16295 y) instructions to conclude the test and establish the necessary post-test state for the TOE.
- 16296 The evaluator prepares for penetration testing based on the list of potential vulnerabilities
16297 identified during the search of the public domain and the analysis of the evaluation evidence.
- 16298 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond
16299 those for which a **High** attack potential is required to effect an attack. However, as a result of
16300 evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only
16301 by an attacker with greater than **High** attack potential. Such vulnerabilities are to be reported in
16302 the ETR as residual vulnerabilities.
- 16303 With an understanding of the potential vulnerability, the evaluator determines the most feasible
16304 way to test for the TOE's susceptibility. Specifically the evaluator considers:
- 16305 the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses (It is
16306 possible that the evaluator will need to use an interface to the TOE other than the TSFI to
16307 demonstrate properties of the TSF such as those described in the security architecture description
16308 (as required by ADV_ARC). It should be noted, that although these TOE interfaces provide a means
16309 of testing the TSF properties, they are not the subject of the test.);
- 16310 initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will
16311 need to exist and security attributes they will need to have);
- 16312 special test equipment that will be required to either stimulate a TSFI or make observations of a
16313 TSFI;
- 16314 whether theoretical analysis should replace physical testing, particularly relevant where the
16315 results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are
16316 likely to succeed after a given number of attempts.
- 16317 The evaluator will probably find it practical to carry out penetration testing using a series of test
16318 cases, where each test case will test for a specific potential vulnerability.
- 16319 The intent of specifying this level of detail in the test documentation is to allow another evaluator
16320 to repeat the tests and obtain an equivalent result.
- 16321 **16.2.5.7.3 Work unit AVA_VAN.5-8**
- 16322 The evaluator ***shall conduct*** penetration testing.
- 16323 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.5-6 as a
16324 basis for executing penetration tests on the TOE, but this does not preclude the evaluator from
16325 performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests
16326 as a result of information learnt during penetration testing that, if performed by the evaluator, are
16327 to be recorded in the penetration test documentation. Such tests may be required to follow up
16328 unexpected results or observations, or to investigate potential vulnerabilities suggested to the
16329 evaluator during the pre-planned testing.
- 16330 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the
16331 evaluator should determine whether or not the evaluator's own analysis was incorrect, or if
16332 evaluation deliverables are incorrect or incomplete.
- 16333 The evaluator is not expected to test for potential vulnerabilities (including those in the public
16334 domain) beyond those which required a **High** attack potential. In some cases, however, it will be
16335 necessary to carry out a test before the exploitability can be determined. Where, as a result of

16336 evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond **High**
 16337 attack potential, this is reported in the ETR as a residual vulnerability.

16338 **16.2.5.7.4 Work unit AVA_VAN.5-9**

16339 The evaluator **shall record** the actual results of the penetration tests.

16340 While some specific details of the actual test results may be different from those expected (e.g. time
 16341 and date fields in an audit record) the overall result should be identical. Any unexpected test
 16342 results should be investigated. The impact on the evaluation should be stated and justified.

16343 **16.2.5.7.5 Work unit AVA_VAN.5-10**

16344 The evaluator **shall report** in the ETR the evaluator penetration testing effort, outlining the testing
 16345 approach, configuration, depth and results.

16346 The penetration testing information reported in the ETR allows the evaluator to convey the overall
 16347 penetration testing approach and effort expended on this sub-activity. The intent of providing this
 16348 information is to give a meaningful overview of the evaluator's penetration testing effort. It is not
 16349 intended that the information regarding penetration testing in the ETR be an exact reproduction of
 16350 specific test steps or results of individual penetration tests. The intention is to provide enough
 16351 detail to allow other evaluators and evaluation authorities to gain some insight about the
 16352 penetration testing approach chosen, amount of penetration testing performed, TOE test
 16353 configurations, and the overall results of the penetration testing activity.

16354 Information that would typically be found in the ETR section regarding evaluator penetration
 16355 testing efforts is:

16356 TOE test configurations. The particular configurations of the TOE that were penetration tested;

16357 TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of
 16358 the penetration testing;

16359 Verdict for the sub-activity. The overall judgement on the results of penetration testing.

16360 This list is by no means exhaustive and is only intended to provide some context as to the type of
 16361 information that should be present in the ETR concerning the penetration testing the evaluator
 16362 performed during the evaluation.

16363

16364 **16.2.5.7.6 Work unit AVA_VAN.5-11**

16365 The evaluator **shall examine** the results of all penetration testing to determine that the TOE, in its
 16366 operational environment, is resistant to an attacker possessing a **High** attack potential.

16367 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by
 16368 an attacker possessing an attack potential less than **or equal to** High, then this evaluator action
 16369 fails.

16370 This text was incorporated from a national scheme document (AIS34 from BSI). References within
 16371 that text to other scheme documents (such as AIS14, 19, 26) have been deleted but additional text
 16372 would be welcome where it might add to clarity

16373 The guidance in B.4 and the guidance for special technical areas that is relevant for the national
 16374 scheme should be used to determine the attack potential required to exploit a particular
 16375 vulnerability and whether it can therefore be exploited in the intended environment. It may not be
 16376 necessary for the attack potential to be calculated in every instance, only if there is some doubt as

16377 to whether or not the vulnerability can be exploited by an attacker possessing an attack potential
16378 less than **or equal to** High.

16379 **16.2.5.7.7 Work unit AVA_VAN.5-12**

16380 The evaluator ***shall report*** in the corresponding ETR-part all exploitable vulnerabilities and
16381 residual vulnerabilities, detailing for each:

16382 z) its source (e.g. ISO/IEC 18045 activity being undertaken when it was conceived, known to
16383 the evaluator, read in a publication);

16384 aa) the SFR(s) not met;

16385 bb) a description;

16386 cc) whether it is exploitable in its operational environment or not (i.e. exploitable or
16387 residual);

16388 dd) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity
16389 and the equipment required to perform the identified vulnerabilities, and the
16390 corresponding values using the tables 3 and 4 of Annex B.4.

16391 **17 Class ACO: Composition**

16392 **17.1 Introduction**

16393 The goal of this activity is to determine whether the components can be integrated in a secure
16394 manner, as defined in the ST for the composed TOE. This is achieved through examination and
16395 testing of the interfaces between the components, supported by examination of the design of the
16396 components and the conduct of vulnerability analysis.

16397 **17.2 Application notes**

16398 The Reliance of dependent component (ACO_REL) family identifies where the dependent
16399 component is reliant upon IT in its operational environment (satisfied by a base component in the
16400 composed TOE evaluation) in order to provide its own security services. This reliance is identified
16401 in terms of the interfaces expected by the dependent component to be provided by the base
16402 component. Development evidence (ACO_DEV) then determines which interfaces of the base
16403 component were considered (as TSFI) during the component evaluation of the base component.

16404 It should be noted that Reliance of dependent component (ACO_REL) does not cover other
16405 evidence that may be needed to address the technical integration problem of composing
16406 components (e.g. descriptions of non-TSF interfaces of the operating system, rules for integration,
16407 etc.). This is outside the security assessment of the composition and is a functional composition
16408 issue.

16409 As part of Composed TOE testing (ACO_CTT) the evaluator will perform testing of the composed
16410 TOE SFRs at the composed TOE interfaces and of the interfaces of the base component relied upon
16411 by the dependent component to confirm they operate as specified. The subset selected will
16412 consider the possible effects of changes to the configuration/use of the base component as used in
16413 the composed TOE. These changes are identified from the configuration of the base component
16414 determined during the base component evaluation. The developer will provide test evidence for
16415 each of the base component interfaces (the requirements for coverage are consistent with those
16416 applied to the evaluation of the base component).

16417 Composition rationale (ACO_COR) requires the evaluator to determine whether the appropriate
16418 assurance measures have been applied to the base component, and whether the base component is

16419 being used in its evaluated configuration. This includes determination of whether all security
 16420 functionality required by the dependent component was within the TSF of the base component.
 16421 The Composition rationale (ACO_COR) requirement may be met through the production of
 16422 evidence that each of these is demonstrated to be upheld. This evidence may be in the form of the
 16423 security target and a public report of the component evaluation (e.g. certification report).

16424 If, on the other hand, one of the above have not been upheld, then it may be possible that an
 16425 argument can be made as to why the assurance gained during an original evaluation is unaffected.
 16426 If this is not possible then additional evaluation evidence for those aspects of the base component
 16427 not covered may have to be provided. This material is then assessed in Development evidence
 16428 (ACO_DEV).

16429 For example, it may be the case as described in the Interactions between entities (see Annex B.3,
 16430 **Interactions between composed IT entities** in ISO/IEC 15408-3) that the dependent component
 16431 requires the base component to provide more security functionality in the composed TOE than
 16432 included in the base component evaluation. This would be determined during the application of the
 16433 Reliance of dependent component (ACO_REL) and Development evidence (ACO_DEV) families. In
 16434 this case the composition rationale evidence provided for Composition rationale (ACO_COR) would
 16435 demonstrate that the assurance gained from the base component evaluation is unaffected. This
 16436 may be achieved by means including:

- 16437 a) Performing a re-evaluation of the base component focusing on the evidence relating to
 16438 the extended part of the TSF;
- 16439 b) Demonstrating that the extended part of the TSF cannot affect other portions of the TSF,
 16440 and providing evidence that the extended part of the TSF provides the necessary security
 16441 functionality.

16442 **17.3 Composition rationale (ACO_COR)**

16443 **17.3.1 Evaluation of sub-activity (ACO_COR.1)**

16444 **17.3.1.1 Input**

16445 The evaluation evidence for this sub-activity is:

- 16446 a) the composed ST;
- 16447 b) the composition rationale;
- 16448 c) the reliance information;
- 16449 d) the development information;
- 16450 e) unique identifier.

16451 **17.3.1.2 Action ACO_COR.1.1E**

16452 ISO/IEC 15408-3 ACO_COR.1.1C: *The composition rationale shall demonstrate that a level of*
 16453 *assurance at least as high as that of the dependent component has been obtained for the support*
 16454 *functionality of the base component, when the base component is configured as required to support*
 16455 *the TSF of the dependent component.*

16456 **17.3.1.2.1 Work unit ACO_COR.1-1**

16457 The evaluator **shall examine** the correspondence analysis with the development information and
 16458 the reliance information to identify the interfaces that are relied upon by the dependent
 16459 component which are not detailed in the development information.

- 16460 The evaluator's goal in this work unit is two fold:
- 16461 a) to determine which interfaces relied upon by the dependent component have had the
16462 appropriate assurance measures applied.
- 16463 b) to determine that the assurance package applied to the base component during the base
16464 component evaluation contained either the same assurance requirements as those in the
16465 package applied to the dependent component during its' evaluation, or hierarchically
16466 higher assurance requirements.
- 16467 The evaluator may use the correspondence tracing in the development information developed
16468 during the Development evidence (ACO_DEV) activities (e.g. ACO_DEV.1-2, ACO_DEV.2-4,
16469 ACO_DEV.3-6) to help identify the interfaces identified in the reliance information that are not
16470 considered in the development information.
- 16471 The evaluator will record the SFR-enforcing interfaces described in the reliance information that
16472 are not included in the development information. These will provide input to ACO_COR.1-3 work
16473 unit, helping to identify the portions of the base component in which further assurance is required.
- 16474 If the both the base and dependent components were evaluated against the same assurance
16475 package, then the determination of whether the level of assurance in the portions within the base
16476 component evaluation is at least as high as that of the dependent component is trivial. If however,
16477 the assurance packages applied to the components during the component evaluations differ, the
16478 evaluator needs to determine that the assurance requirements applied to the base component are
16479 all hierarchically higher to the assurance requirements applied to the dependent component.
- 16480 **17.3.1.2.2 Work unit ACO_COR.1-2**
- 16481 The evaluator *shall examine* the composition rationale to determine, for those included base
16482 component interfaces on which the dependent TSF relies, whether the interface was considered
16483 during the evaluation of the base component.
- 16484 The ST, component public evaluation report (e.g. certification report) and guidance documents for
16485 the base component all provide information on the scope and boundary of the base component.
16486 The ST provides details of the logical scope and boundary of the composed TOE, allowing the
16487 evaluator to determine whether an interface relates to a portion of the product that was within the
16488 scope of the evaluation. The guidance documentation provides details of use of all interfaces for the
16489 composed TOE. Although the guidance documentation may include details of interfaces in the
16490 product that are not within the scope of the evaluation, any such interfaces should be identifiable,
16491 either from the scoping information in the ST or through a portion of the guidance that deals with
16492 the evaluated configuration. The public evaluation report may provide any additional constraints
16493 on the use of the composed TOE that are necessary.
- 16494 Therefore, the combination of these inputs allows the evaluator to determine whether an interface
16495 described in the composition rationale has the necessary assurance associated with it, or whether
16496 further assurance is required. The evaluator will record those interfaces of the base component for
16497 which additional assurance is required, for consideration during ACO_COR.1-3.
- 16498 **17.3.1.2.3 Work unit ACO_COR.1-3**
- 16499 The evaluator *shall examine* the composition rationale to determine that the necessary assurance
16500 measures have been applied to the base component.
- 16501 The evaluation verdicts, and resultant assurance, for the base component can be reused provided
16502 the same portions of the base component are used in the composed TOE and they are used in a
16503 consistent manner.

- 16504 In order to determine whether the necessary assurance measures have already been applied to the
 16505 component, and the portions of the component for which assurance measures still need to be
 16506 applied, the evaluator should use the output of the ACO_DEV.*.2E action and the work units
 16507 ACO_COR.1-1 and ACO_COR.1-2:
- 16508 a) For those interfaces identified in the reliance information (Reliance of dependent
 16509 component (ACO_REL)), but not discussed in development information (Development
 16510 evidence (ACO_DEV)), additional information is required. (Identified in ACO_COR.1-1.)
- 16511 b) For those interfaces used inconsistently in the composed TOE from the base component
 16512 (difference between the information provided in Development evidence (ACO_DEV) and
 16513 Reliance of dependent component (ACO_REL) the impact of the differences in use need to
 16514 be considered. (Identified in ACO_DEV.*.2E.)
- 16515 c) For those interfaces identified in composition rationale for which no assurance has
 16516 previously been gained, additional information is required. (Identified in ACO_COR.1-2.)
- 16517 d) For those interfaces consistently described in the reliance information, composition
 16518 rationale and the development information, no further action is required as the results
 16519 from the base component evaluation can be re-used.
- 16520 The interfaces of the base component reported to be required by the reliance information but not
 16521 included in the development information indicate the portions of the base component where
 16522 further assurance is required. The interfaces identify the entry points into the base component.
- 16523 For those interfaces included in both the development information and reliance information, the
 16524 evaluator is to determine whether the interfaces are being used in the composed TOE in a manner
 16525 that is consistent with the base component evaluation. The method of use of the interface will be
 16526 considered during the Development evidence (ACO_DEV) activities to determine that the use of the
 16527 interface is consistent in both the base component and the composed TOE. The remaining
 16528 consideration is the determination of whether the configurations of the base component and the
 16529 composed TOE are consistent. To determine this, the evaluator will consider the guidance
 16530 documentation of each to ensure they are consistent (see further guidance below regarding
 16531 consistent guidance documentation). Any deviation in the documentation will be further analysed
 16532 by the evaluation to determine the possible effects.
- 16533 For those interfaces that are consistently described in the reliance information and development
 16534 information, and for which the guidance is consistent for the base component and the composed
 16535 TOE, the required level of assurance has been provided.
- 16536 The following subsubclauses provide guidance on how to determine consistency between
 16537 assurance gained in the base component, the evidence provided for the composed TOE, and the
 16538 analysis performed by the evaluator in the instances where inconsistencies are identified.
- 16539 **17.3.1.2.3.1 Development**
- 16540 The reliance information identifies the interfaces in the dependent component that are to be
 16541 matched by the base component. If an interface identified in the reliance information is not
 16542 identified in the development information, then the composition rationale is to provide a
 16543 justification of how the base component provides the required interfaces.
- 16544 If an interface identified in the reliance information is identified in the development information,
 16545 but there are inconsistencies between the descriptions, further analysis is required. The evaluator
 16546 identifies the differences in use of the base component as considered in the base component
 16547 evaluation and the composed TOE evaluation. The evaluator will devise testing to be performed
 16548 (during the conduct of Composed TOE testing (ACO_CTT)) to test the interface.

The patch status of the base and dependent components as used in the composed TOE should be compared to the patch status of the components during the component evaluations. If any patches have been applied to the components, the composition rationale is to include details of the patches, including any potential impact to the SFRs of the evaluated component. The evaluator should consider the details of the changes provided and verify the accuracy of the potential impact of the change on the component SFRs. The evaluator should then consider whether the changes made by the patch should be verified through testing, and will identify the necessary testing approach. The testing may take the form of repeating the applicable evaluator/developer testing performed for the component evaluation of the component or it may be necessary for the evaluator to devise new tests to confirm the modified component.

If any of the individual components have been the subject of assurance continuity activities since the completion of the component evaluation, the evaluator will consider the changes assessed in the assurance continuity activities during the independent vulnerability analysis activity for the composed TOE (in Composition vulnerability analysis (ACO_VUL)).

17.3.1.2.3.2 Guidance

The guidance for the composed TOE is likely to make substantial reference out to the guidance for the individual components. The minimal guidance expected to be necessary is the identification of any ordering dependencies in the application of guidance for the dependent and base components, particularly during the preparation (installation) of the composed TOE.

In addition to the application of the Preparative procedures (AGD_PRE) and Operational user guidance (AGD_OPE) families to the guidance for the composed TOE, it is necessary to analyse the consistency between the guidance for the components and the composed TOE, to identify any deviations.

If the composed TOE guidance refers out to the base component and dependent component guidance, then the consideration for consistency is limited to consistency between the guidance documentation provided for each of the components (i.e. consistency between the base component guidance and the dependent component guidance). However, if additional guidance is provided for the composed TOE, to that provided for the components, greater analysis is required, as consistency is also required between the guidance documentation for the components and guidance documentation for the composed TOE.

Consistent in this instance is understood to mean that either the guidance is the same or it places additional constraints on the operation of the individual components when combined, in a similar manner to *refinement* of functional/assurance components.

17.3.1.3 With the information available (that used as input for Development evidence (ACO_DEV) or the development aspects discussed above) the evaluator may be able to determine all possible impacts of the deviation from the configuration of the base component specified in the component evaluation. However, for high EALs (where evaluation of the base component included Objectives

17.3.1.3 The objectives of this sub-activity are to determine whether the formal security policy model of the TSF clearly and consistently describes the rules and characteristics of the security policies and whether this description corresponds with the description of security functions in the functional specification.

17.3.1.3 Input

17.3.1.3 The evaluation evidence for this sub-activity is:

17.3.1.3 the ST;

17.3.1.3 the functional specification;

- 16595 **17.3.1.3** formal security policy model (ADV_SPM.1.1D);
- 16596 **17.3.1.3** formal proof of correspondence between the model and any formal functional
16597 specification (ADV_SPM.1.3D);
- 16598 **17.3.1.3** demonstration of correspondence between the model and the functional specification
16599 (ADV_SPM.1.4D).
- 16600 **17.3.1.3 Application notes**
- 16601 **17.3.1.3** This activity applies to cases where the developer has provided a formal security policy
16602 model of the TOE.
- 16603 **17.3.1.3** A formal TOE security policy model is a representation of the rules (synonymously
16604 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
16605 terms. Their formal counterparts are called security properties and security features,
16606 respectively. The representation includes but is not limited to algebraic specifications, finite state
16607 machines and logic formalisms strong enough to formally infer the properties from the features.
16608 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
16609 characteristics are mapped to the respective properties and features.
- 16610 **17.3.1.3** The creation of a formal security policy model helps to identify and eliminate
16611 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
16612 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
16613 judgement of how well the developer has understood the security functionality being
16614 implemented and whether there are inconsistencies between the security requirements and the
16615 TOE design. The confidence in the model is accompanied by a proof that it contains no
16616 inconsistencies.
- 16617 **17.3.1.3** A formal security model is a precise formal presentation of the important aspects of
16618 security and their relationship to the behaviour of the TOE; it identifies the set of rules
16619 (principles) that defines the TOE security policy and the set of practises (characteristics) that
16620 regulates how the TSF manages, protects, and otherwise controls the system resources. The
16621 model includes the set of restrictions and properties that specify how information and computing
16622 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
16623 engineering arguments showing that these restrictions and properties play a key role in the
16624 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
16625 as well as ancillary text to explain the model and to provide it with context. The security
16626 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
16627 with the rest of the TOE and with its operational environment), as well as its internal behaviour.
- 16628 **17.3.1.3** The Security Policy Model of the TOE is informally abstracted from its realisation by
16629 considering the proposed security requirements of the ST. The informal abstraction is taken to be
16630 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
16631 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
16632 are always prone to fallacies; especially if relationships among subjects, objects and operations
16633 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
16634 characteristics of the security policy model are mapped to respective properties and features
16635 within some formal system, whose rigour and strength can afterwards be used to obtain the
16636 security properties by means of theorems and formal proof.
- 16637 **17.3.1.3** While the term “formal security policy model” is used in academic circles, the CC's
16638 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
16639 claimed. Therefore, the formal security policy model is merely a formal representation of the set
16640 of SFRs being claimed.
- 16641 **17.3.1.3** The term security policy has traditionally been associated with only access control
16642 policies, whether label-based (mandatory access control) or user-based (discretionary access

16643	control). However, a security policy is not limited to access control; there are also audit policies,
16644	identification policies, authentication policies, encryption policies, management policies, and any
16645	other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
16646	contains an assignment for identifying these policies that are formally modelled.
16647	17.3.1.3 It is recognized that not all policies can be formally modelled for all TOEs. This is
16648	because either a given policy can not be formally modelled in the otherwise well suited
16649	framework, or because the nature of the TOE renders impossible the modelling of policies that
16650	would otherwise be possible to model.
16651	17.3.1.3 Action ADV_SPM.1.1E
16652	17.3.1.3 ADV_SPM.1.1C <i>The model shall be in a formal style, supported by explanatory</i>
16653	<i>text as required, and identify the security policies of the TSF that are modelled.</i>
16654	17.3.1.3 Work unit ADV_SPM.1-1
16655	17.3.1.3 The evaluator <i>shall examine the TOE security policy model to determine that it is</i>
16656	written in a formal style.
16657	17.3.1.3 The evaluator identifies the formal framework upon which the TOE security policy
16658	model is based and ensures that it is founded on well established mathematical concepts. They
16659	also identify the security properties and features addressed in the application notes and ensure
16660	the formalization of at least one security policy.
16661	17.3.1.3 For guidance on formal methods refer to ISO/IEC 15408-3
16662	17.3.1.3 Work unit ADV_SPM.1-2
16663	17.3.1.3 The evaluator <i>shall examine the TOE security policy model to determine that it</i>
16664	contains all necessary informal explanatory text.
16665	17.3.1.3 Supporting narrative descriptions are necessary for all parts of the model (for example,
16666	to make clear the meaning of any formal notation and how they are used) including the security
16667	properties and features.
16668	17.3.1.3 Work unit ADV_SPM.1-3
16669	17.3.1.3 The evaluator <i>shall examine the TOE security policy model to determine that all</i>
16670	security policies of the TSF are identified that are modelled.
16671	17.3.1.3 The evaluator determines whether the SPM identifies the security policies for which a
16672	model is provided, identifying the relevant portions of the statement of SFRs that comprise each
16673	of the modelled policies.
16674	17.3.1.3 The evaluator determines whether the list of security policies identified by the SPM is
16675	consistent with the assignment of ADV_SPM.1.1D in the ST.
16676	17.3.1.3 The evaluator determines whether for each security policy identified by the SPM a
16677	model is in fact provided.
16678	17.3.1.3 ADV_SPM.1.2C <i>For all policies that are modelled, the model shall define</i>
16679	<i>security for the TOE and provide a formal proof that the TOE cannot reach a state that is</i>
16680	<i>not secure.</i>

- 16681 **17.3.1.3 Work unit ADV_SPM.1-4**
- 16682 **17.3.1.3** The evaluator *shall examine the principles and characteristics of the security*
 16683 policies to determine that the modelled security behaviour of the TOE is clearly articulated.
- 16684 **17.3.1.3** The security policies are expressed in terms of security principles (rules) which are
 16685 modelled by security properties and define the secure state of the TOE. For example, a model
 16686 based on state transitions could describe the security policies in terms of principles of its states,
 16687 identify its initial state, and define what it means to be a secure state.
- 16688 **17.3.1.3** The evaluator determines that the security policies are reflected within their formal
 16689 counterparts of the TSP model.
- 16690 **17.3.1.3** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 16691 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 16692 resources including attributes and conditions of the TOE) which are modelled by security
 16693 features. For example, a model based on state transitions could describe the characteristics as
 16694 possible actions in each secure state in a level of detail sufficient to decide into which state the
 16695 TOE will be transformed by that action.
- 16696 **17.3.1.3** Together the security principles and characteristics describe the entire security posture
 16697 of the TOE.
- 16698 **17.3.1.3** In the context of a formal TOE security policy model the security behaviour is
 16699 considered to be clearly articulated only if an adequate mapping from principles and
 16700 characteristics to their respective formal counterparts properties and features has been given.
 16701 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 16702 detailed enough to allow for correct identification of all security objectives and the relation to the
 16703 security environment.
- 16704 **17.3.1.3** The above condition for clear articulation is necessary but not sufficient. An informal
 16705 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 16706 must be provided in order to make clear their intended meaning.
- 16707 **17.3.1.3 Work unit ADV_SPM.1-5**
- 16708 **17.3.1.3** The evaluator *shall examine the TOE security policy model rationale to determine that*
 16709 it formally proves that the security features enforce the security properties.
- 16710 **17.3.1.3** To determine the enforcement, the evaluator considers the security properties and the
 16711 security features and verifies that the arguments used in the proof are valid. The proof of
 16712 correspondence between the security properties and the security features shall be formal.
- 16713 **17.3.1.3** The validity of the security properties shall mean that the TOE is in a secure state. By
 16714 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 16715 state.
- 16716 **17.3.1.3 Work unit ADV_SPM.1-6**
- 16717 **17.3.1.3** The evaluator *shall examine the TOE security policy model rationale to determine that*
 16718 it proves the internal consistency of the TOE security policy model.
- 16719 **17.3.1.3** The proof shall show the absence of contradictions within the TOE security policy
 16720 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 16721 used in the proof are valid.
- 16722 **17.3.1.3** Since the TOE security policy model is formal, the proof of its internal consistency shall
 16723 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE

16724	security policy model usually is not possible due to the fundamental nature of formal frameworks.
16725	Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
16726	security policy model that prove the internal consistency by means of a combination with generic
16727	arguments of the formal framework.
16728	17.3.1.3 ADV_SPM.1.3C <i>The correspondence between the model and the functional</i>
16729	<i>specification shall be at the correct level of formality.</i>
16730	17.3.1.3 Work unit ADV_SPM.1-7
16731	17.3.1.3 The evaluator <i>shall examine the correspondence between the model and the functional</i>
16732	specification to determine that a semiformal demonstration of correspondence between the
16733	model and any semiformal functional specification is provided.
16734	17.3.1.3 This work unit is only applicable to a semiformal presentation of the functional
16735	specification, which is required by ADV_FSP.5.2C.
16736	17.3.1.3 A semiformal correspondence is one that results from a structured approach with a
16737	substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
16738	mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
16739	terms, and so it provides less ambiguity than would exist in an informal correspondence.
16740	17.3.1.3 For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
16741	methods’.
16742	17.3.1.3 Work unit ADV_SPM.1-8
16743	17.3.1.3 The evaluator <i>shall examine the correspondence between the model and the functional</i>
16744	specification to determine that a formal proof of correspondence between the model and any
16745	formal functional specification is provided.
16746	17.3.1.3 This work unit is only applicable to a formal presentation of the functional specification,
16747	which is required by ADV_FSP.6.2D.
16748	17.3.1.3 There should be a formal proof of correspondence between the model and any formal
16749	functional specification.
16750	17.3.1.3 The formal proof of correspondence removes all subjective interpretations of its terms
16751	by enlisting well-established mathematical concepts to define the syntax and semantics of the
16752	formal notation and uses rules that support logical reasoning. The security features within the
16753	TOE (which are identified in the formal TSP model) are expressed in a formal specification
16754	language and shown to be satisfied by the formal specification.
16755	17.3.1.3 For guidance on formal methods refer to ISO/IEC 15408-3 .
16756	17.3.1.3 ADV_SPM.1.4C <i>The correspondence shall show that the functional</i>
16757	<i>specification is consistent and complete with respect to the model.</i>
16758	17.3.1.3 Work unit ADV_SPM.1-9
16759	17.3.1.3 The evaluator <i>shall examine the correspondence to determine that the behaviour at the</i>
16760	TSF interfaces (as articulated in the functional specification) is complete with respect to the
16761	behaviour modelled by the security features.
16762	17.3.1.3 The term “correspondence” here means both the formal proof of correspondence
16763	between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
16764	of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.

- 16765 **17.3.1.3** In determining completeness of the correspondence, the evaluator considers the
 16766 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 16767 features of the TSP model. The demonstration should show that all characteristics belonging to
 16768 policies that are required to be modelled have an associated feature description in the TOE
 16769 security policy model, and that each feature of the TSP model does occur in the mapping.
- 16770 **17.3.1.3** Abstention from formally modelling TSFI behaviour always calls for justification on the
 16771 developer's side (also confer the application notes above).
- 16772 **17.3.1.3 Work unit ADV_SPM.1-10**
- 16773 **17.3.1.3** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 16774 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 16775 behaviour modelled by the security features.
- 16776 **17.3.1.3** The term "correspondence" here means both the formal proof of correspondence
 16777 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 16778 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 16779 **17.3.1.3** The meaning of consistency reflects the conventional understanding in contrast to the
 16780 internal consistency concept of work unit ADV_SPM.1-6.
- 16781 **17.3.1.3** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 16782 security features established in the preceding work unit and verifies that the correspondence
 16783 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 16784 behaviour.
- 16785 **17.3.1.3** For example, if TSFI behaviour dealt with access management on the granularity of
 16786 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 16787 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 16788 management for groups of users, then a TSP model describing the security behaviour of the TOE
 16789 in terms of individual users would also not be consistent.
- 16790 **17.3.1.3** As another example, if remote untrusted users had to pass more stringent
 16791 authentication procedures than administrators whose only point of access were within a
 16792 physically-protected area, then this difference in authentication procedures had to be reflected in
 16793 the security features.
- 16794 **17.3.1.3** TOE design (ADV_TDS) requirements) it is possible that, unless detailed design
 16795 abstractions for the base component are delivered as part of the development information for the
 16796 composed TOE, the possible impacts of the modification to the guidance cannot be fully determined
 16797 as the internals are unknown. In this case the evaluator will report the residual risk of the analysis.
- 16798 These residual risks are to be included in any public evaluation report for the composed TOE.
- 16799 The evaluator will note these variances in the guidance for input into evaluator independent
 16800 testing activities (Composed TOE testing (ACO_CTT)).
- 16801 The guidance for the composed TOE may add to the guidance for the components, particularly in
 16802 terms of installation and the ordering of installation steps for the base component in relation to the
 16803 installation steps for the dependent component. The ordering of the steps for the installation of the
 16804 individual components should not change, however they may need to be interleaved. The evaluator
 16805 will examine this guidance to ensure that it still meets the requirement of the AGD_PRE activity
 16806 performed during the evaluations of the components.
- 16807 It may be the case that the reliance information identifies that interfaces of the base component, in
 16808 addition to those identified as TSFIs of the base component, are relied upon by the dependent
 16809 component are identified in the reliance information. It may be necessary for guidance to be

provided for the use of any such additional interfaces in the base component. Provided the consumer of the composed TOE is to receive the guidance documentation for the base component, then the results of the AGD_PRE and AGD_OPE verdicts for the base component can be reused for those interfaces considered in the evaluation of the base component. However, for the additional interfaces relied upon by the dependent component, the evaluator will need to determine that the guidance documentation for the base component meets the requirements of AGD_PRE and AGD_OPE, as applied in the base component evaluations.

For those interfaces considered during the base component evaluation, and therefore, for which assurance has already been gained, the evaluator will ensure that the guidance for the use of each interface for the composed TOE is consistent with that provided for the base component. To determine the guidance for the composed TOE is consistent with that for the base component, the evaluator should perform a mapping for each interface to the guidance provided for both the composed TOE and the base component. The evaluator then compares the guidance to determine consistency.

Examples of additional constraints provided in composed TOE guidance that would be considered to be consistent with component guidance are (guidance for a component is given followed by an example of guidance for a composed TOE that would be considered to provide additional constraints):

— Component: The password length must be set to a minimum of 8 characters length, including alphabetic and numeric characters.

— Composed TOE: The password length must be set to a minimum of 10 characters in length, including alphabetic and numeric characters and *at least one of the following special characters: () {} ^ < > - _*

— NOTE: It would only be acceptable to increase the password length to [*integer* > 8] characters while removing the mandate for the inclusion of both alphabetic and numeric characters for the composed TOE, if the same or a higher metric was achieved for the strength rating (taking into account the likelihood of the password being guessed).

— Component: The following services are to be disabled in the registry settings: WWW Publishing Service and ICDBReporter service.

— Composed TOE: The following services are to be disabled in the registry settings: Publishing Service, ICDBReporter service, Remote Procedure Call (RPC) Locator and Procedure Call (RPC) Service.

— Component: Select the following attributes to be included in the accounting log files: date, time, type of event, subject identity and success/failure.

— Composed TOE: Select the following attributes to be included in the accounting log files: date, time, type of event, subject identity, success/failure, *event message and process thread*.

If the guidance for the composed TOE deviates (is not a refinement) from that provided for the base component, the evaluator will assess the potential risks of the modification to the guidance. The evaluator will use the information available (including that provided in the public domain, the architectural description of the base component in the public evaluation report (e.g. certification report), the context of the guidance from the remainder of the guidance documentation) to identify likely impact of the modification to the guidance on the SFRs of the composed TOE.

If during the dependent component evaluation the trial installation used the base component to satisfy the environment requirements of the dependent component this work unit for the composed TOE is considered to be satisfied. If the base component was not used in satisfaction of the work unit AGD_PRE.1-3 during the dependent component evaluation, the evaluator will apply

16856 the user procedures provided for the composed TOE to prepare the composed TOE, in accordance
 16857 with the guidance specified in AGD_PRE.1-3. This will allow the evaluator to determine that the
 16858 preparative guidance provided for the composed TOE is sufficient to prepare the composed TOE
 16859 and its operational environment securely.

16860 **17.3.1.6.10.1 Life-cycle**

16861 **Delivery**

16862 If there is a different delivery mechanism used for the delivery of the composed TOE (i.e. the
 16863 components are not delivered to the consumer in accordance with the secure delivery procedures
 16864 defined and assessed during the evaluation of the components), the delivery procedures for the
 16865 composed TOE will require evaluation against the Delivery (ALC_DEL) requirements applied
 16866 during the components evaluations.

16867 The composed TOE may be delivered as an integrated product or may require the components to
 16868 be delivered separately.

16869 If the components are delivered separately, the results of the delivery of the base component and
 16870 dependent component are reused. The delivery of the base component is checked during the
 16871 evaluator trial installation of the dependent component, using the specified guidance and checking
 16872 the aspects of delivery that are the responsibility of the user, as described in the guidance
 16873 documentation for the base component.

16874 If the composed TOE is delivered as a new entity, then the method of delivery of that entity must be
 16875 considered in the composed TOE evaluation activities.

16876 The assessment of the delivery procedures for composed TOE items is to be performed in
 16877 accordance with the methodology for Delivery (ALC_DEL) as for any other [component] TOE,
 16878 ensuring any additional items (e.g. additional guidance documents for the composed TOE) are
 16879 considered in the delivery procedures.

16880 **CM Capabilities**

16881 The unique identification of the composed TOE is considered during the application of Evaluation
 16882 of sub-activity (ALC_CMC.1) and the items from which that composed TOE is comprised are
 16883 considered during the application of Evaluation of sub-activity (ALC_CMS.2).

16884 Although additional guidance may be produced for the composed TOE, the unique identification of
 16885 this guidance (considered as part of the unique identification of the composed TOE during
 16886 Evaluation of sub-activity (ALC_CMC.1)) is considered sufficient control of the guidance.

16887 The verdicts of the remaining (not considered above) Class ALC: Life-cycle support activities can be
 16888 reused from the base component evaluation, as no further development is performed during
 16889 integration of the composed TOE.

16890 There are no additional considerations for development security as the integration is assumed to
 16891 take place at either the consumer's site or, in the instance that the composed TOE is delivered as an
 16892 integrated product, at the site of the dependent component developer. Control at the consumer's
 16893 site is outside the consideration of ISO/IEC 15408. No additional requirements or guidance are
 16894 necessary if integration is at the same site as that for the dependent component, as all components
 16895 are considered to be configuration items for the composed TOE, and should therefore be
 16896 considered under the dependent component developer's security procedures anyway.

16897 Tools and techniques adopted during integration will be considered in the evidence provided by
 16898 the dependent component developer. Any tools/techniques relevant to the base component will
 16899 have been considered during the evaluation of the base component. For example, if the base
 16900 component is delivered as source code and requires compilation by the consumer (e.g. dependent

16901 component developer who is performing integration) the compiler would have been specified and
16902 assessed, along with the appropriate arguments, during evaluation of the base component.

16903 There is no life-cycle definition applicable to the composed TOE, as no further development of
16904 items is taking place.

16905 The results of flaw remediation for a component are not applicable to the composed TOE. If flaw
16906 remediation is included in the assurance package for the composed TOE, then the Flaw
16907 remediation (ALC_FLR) requirements are to be applied during the composed TOE evaluation (as
16908 for any augmentation).

16909 **17.3.1.6.10.2 Tests**

16910 The composed TOE will have been tested during the conduct of the Class ATE: Tests activities for
16911 evaluation of the dependent component, as the configurations used for testing of the dependent
16912 component should have included the base component to satisfy the requirements for IT in the
16913 operational environment. If the base component was not used in the testing of the dependent
16914 component for the dependent component evaluation, or the configuration of either component
16915 varied from their evaluated configurations, then the developer testing performed for evaluation of
16916 the dependent component to satisfy the Class ATE: Tests requirements is to be repeated on the
16917 composed TOE.

16918 **17.4 Development evidence (ACO_DEV)**

16919 **17.4.1 Evaluation of sub-activity (ACO_DEV.1)**

16920 **17.4.1.1 Objectives**

16921 The objective of this sub-activity is to determine that the appropriate security functionality is
16922 provided by the base component to support the dependent component. This is achieved through
16923 examination of the interfaces of the base component to determine that they are consistent with the
16924 interfaces specified in the reliance information; those required by the dependent component.

16925 The description of the interfaces into the base component is to be provided at a level of detail
16926 consistent with Evaluation of sub-activity (ADV_FSP.2) although not all of the aspects necessary for
16927 satisfaction of Evaluation of sub-activity (ADV_FSP.2) are required for Evaluation of sub-activity
16928 (ACO_DEV.1), as once the interface has been identified and the purpose described the remaining
16929 detail of the interface specification can be reused from evaluation of the base component.

16930 **17.4.1.2 Input**

16931 The evaluation evidence for this sub-activity is:

- 16932 a) the composed ST;
- 16933 b) the development information;
- 16934 c) the reliance information.

16935 **17.4.1.3 Action ACO_DEV.1.1E**

16936 ISO/IEC 15408-3 ACO_DEV.1.1C: *The development information shall describe the purpose of each*
16937 *interface of the base component used in the composed TOE.*

16938 **17.4.1.3.1 Work unit ACO_DEV.1-1**

16939 The evaluator ***shall examine*** the development information to determine that it describes the
16940 purpose of each interface.

16941 The base component provides interfaces to support interaction with the dependent component in
 16942 the provision of the dependent TSF. The purpose of each interface is to be described at the same
 16943 level as the description of the interfaces to the dependent component TSF functionality, as would
 16944 be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This
 16945 description is to provide the reader with an understanding of how the base component provides
 16946 the services required by the dependent component TSF.

16947 This work unit may be satisfied by the provision of the functional specification for the base
 16948 component for those interfaces that are TSFIs of the base component.

16949 ISO/IEC 15408-3 ACO_DEV.1.2C: *The development information shall show correspondence between*
 16950 *the interfaces, used in the composed TOE, of the base component and the dependent component to*
 16951 *support the TSF of the dependent component.*

16952 **17.4.1.3.2 Work unit ACO_DEV.1-2**

16953 The evaluator ***shall examine*** the development information to determine the correspondence,
 16954 between the interfaces of the base component and the interfaces on which the dependent
 16955 component relies, is accurate.

16956 The correspondence between the interfaces of the base component and the interfaces on which the
 16957 dependent component relies may take the form of a matrix or table. The interfaces that are relied
 16958 upon by the dependent component are identified in the reliance information (as examined during
 16959 Reliance of dependent component (ACO_REL) activity).

16960 There is, during this activity, no requirement to determine completeness of the coverage of
 16961 interfaces that are relied upon by the dependent component, only that the correspondence is
 16962 correct and ensuring that interfaces of the base component are mapped to interfaces required by
 16963 the dependent component wherever possible. The completeness of the coverage is considered in
 16964 Composition rationale (ACO_COR) activities.

16965 **17.4.1.4 Action ACO_DEV.1.2E**

16966 **17.4.1.4.1 Work unit ACO_DEV.1-3**

16967 The evaluator ***shall examine*** the development information and the reliance information to
 16968 determine that the interfaces are described consistently.

16969 The evaluator's goal in this work unit is to determine that the interfaces described in the
 16970 development information for the base component and the reliance information for the dependent
 16971 component are represented consistently.

16972 **17.4.2 Evaluation of sub-activity (ACO_DEV.2)**

16973 **17.4.2.1 Objectives**

16974 The objective of this sub-activity is to determine that the appropriate security functionality is
 16975 provided by the base component to support the dependent component. This is achieved through
 16976 examination of the interfaces and associated security behaviour of the base component to
 16977 determine that they are consistent with the interfaces specified in the reliance information; those
 16978 required by the dependent component.

16979 **17.4.2.2 Input**

16980 The evaluation evidence for this sub-activity is:

16981 a) the composed ST;

16982 b) the development information;

16983 c) reliance information.

16984 **17.4.2.3 Action ACO_DEV.2.1E**

16985 ISO/IEC 15408-3 ACO_DEV.2.1C: *The development information shall describe the purpose and*
16986 *method of use of each interface of the base component used in the composed TOE.*

16987 **17.4.2.3.1 Work unit ACO_DEV.2-1**

16988 The evaluator ***shall examine*** the development information to determine that it describes the
16989 purpose of each interface.

16990 The base component provides interfaces to support interaction with the dependent component in
16991 the provision of the dependent TSF. The purpose of each interface is to be described at the same
16992 level as the description of the interfaces to the dependent component TSF functionality, as would
16993 be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This
16994 description is to provide the reader with an understanding of how the base component provides
16995 the services required by the dependent component TSF.

16996 This work unit may be satisfied by the provision of the functional specification for the base
16997 component for those interfaces that are TSFIs of the base component.

16998 **17.4.2.3.2 Work unit ACO_DEV.2-2**

16999 The evaluator ***shall examine*** the development information to determine that it describes the
17000 method of use for each interface.

17001 The method of use for an interface summarises how the interface is manipulated in order to invoke
17002 the operations and obtain results associated with the interface. The evaluator should be able to
17003 determine from reading this material in the development information how to use each interface.
17004 This does not necessarily mean that there needs to be a separate method of use for each interface,
17005 as it may be possible to describe in general how APIs are invoked, for instance, and then identify
17006 each interface using that general style.

17007 This work unit may be satisfied by the provision of the functional specification for the base
17008 component for those interfaces that are TSFIs of the base component.

17009 ISO/IEC 15408-3 ACO_DEV.2.2C: *The development information shall provide a high-level description*
17010 *of the behaviour of the base component, which supports the enforcement of the dependent component*
17011 *SFRs.*

17012 **17.4.2.3.3 Work unit ACO_DEV.2-3**

17013 The evaluator ***shall examine*** the development information to determine that it describes the
17014 behaviour of the base component that supports the enforcement of the dependent component SFRs.

17015 The dependent component invokes interfaces of the base component for the provision of services
17016 by the base component. For the interfaces of the base component that are invoked, the
17017 development information shall provide a high-level description of the associated security
17018 behaviour of the base component. The description of the base component security behaviour will
17019 outline how the base component provides the necessary service when the call to the interface is
17020 made. This description is to be at a level similar to that provided for ADV_TDS.1.4C. Therefore, the
17021 provision of the TOE design evidence from the base component evaluation would satisfy this work
17022 unit, where the interfaces invoked by the dependent component are TSFI of the base component. If
17023 the interfaces invoked by the dependent component are not TSFIs of the base component it is the

17024 associated security behaviour will not necessarily be described in the base component TOE design
17025 evidence.

17026 ISO/IEC 15408-3 ACO_DEV.2.3C: *The development information shall show correspondence between*
17027 *the interfaces, used in the composed TOE, of the base component and the dependent component to*
17028 *support the TSF of the dependent component.*

17029 **17.4.2.3.4 Work unit ACO_DEV.2-4**

17030 The evaluator ***shall examine*** the development information to determine the correspondence,
17031 between the interfaces of the base component and the interfaces on which the dependent
17032 component relies, is accurate.

17033 The correspondence between the interfaces of the base component and the interfaces on which the
17034 dependent component relies may take the form of a matrix or table. The interfaces that are relied
17035 upon by the dependent component are identified in the reliance information (as examined during
17036 Reliance of dependent component (ACO_REL)).

17037 There is, during this activity, no requirement to determine completeness of the coverage of
17038 interfaces that are relied upon by the dependent component, only that the correspondence is
17039 correct and ensuring that interfaces of the base component are mapped to interfaces required by
17040 the dependent component wherever possible. The completeness of the coverage is considered in
17041 Composition rationale (ACO_COR) activities.

17042 **17.4.2.4 Action ACO_DEV.2.2E**

17043 **17.4.2.4.1 Work unit ACO_DEV.2-5**

17044 The evaluator ***shall examine*** the development information and the reliance information to
17045 determine that the interfaces are described consistently.

17046 The evaluator's goal in this work unit is to determine that the interfaces described in the
17047 development information for the base component and the reliance information for the dependent
17048 component are represented consistently.

17049 **17.4.3 Evaluation of sub-activity (ACO_DEV.3)**

17050 **17.4.3.1 Objectives**

17051 The objective of this sub-activity is to determine that the appropriate security functionality is
17052 provided by the base component to support the dependent component. This is achieved through
17053 examination of the interfaces and associated security behaviour of the base component to
17054 determine that they are consistent with the interfaces specified in the reliance information; those
17055 required by the dependent component.

17056 In addition to the interface description, the subsystems of the base component that provide the
17057 security functionality required by the dependent component will be described to enable the
17058 evaluator to determine whether or not that interface formed part of the TSF of the base component.

17059 **17.4.3.2 Input**

17060 The evaluation evidence for this sub-activity is:

- 17061 a) the composed ST;
- 17062 b) the development information;
- 17063 c) reliance information.

17064 **17.4.3.3 Action ACO_DEV.3.1E**

17065 ISO/IEC 15408-3 ACO_DEV.3.1C: *The development information shall describe the purpose and*
17066 *method of use of each interface of the base component used in the composed TOE.*

17067 **17.4.3.3.1 Work unit ACO_DEV.3-1**

17068 The evaluator ***shall examine*** the development information to determine that it describes the
17069 purpose of each interface.

17070 The base component provides interfaces to support interaction with the dependent component in
17071 the provision of the dependent TSF. The purpose of each interface is to be described at the same
17072 level as the description of the interfaces to the dependent component TSF functionality, as would
17073 be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This
17074 description is to provide the reader with an understanding of how the base component provides
17075 the services required by the dependent component TSF.

17076 This work unit may be satisfied by the provision of the functional specification for the base
17077 component for those interfaces that are TSFIs of the base component.

17078 **17.4.3.3.2 Work unit ACO_DEV.3-2**

17079 The evaluator ***shall examine*** the development information to determine that it describes the
17080 method of use for each interface.

17081 The method of use for an interface summarises how the interface is manipulated in order to invoke
17082 the operations and obtain results associated with the interface. The evaluator should be able to
17083 determine from reading this material in the development information how to use each interface.
17084 This does not necessarily mean that there needs to be a separate method of use for each interface,
17085 as it may be possible to describe in general how APIs are invoked, for instance, and then identify
17086 each interface using that general style.

17087 This work unit may be satisfied by the provision of the functional specification for the base
17088 component for those interfaces that are TSFIs of the base component.

17089 ISO/IEC 15408-3 ACO_DEV.3.2C: *The development information shall identify the subsystems of the*
17090 *base component that provide interfaces of the base component used in the composed TOE.*

17091 **17.4.3.3.3 Work unit ACO_DEV.3-3**

17092 The evaluator ***shall examine*** the development information to determine that all subsystems of the
17093 base component that provide interfaces to the dependent component are identified.

17094 **17.4.3.4 For those interfaces that are considered to form part of the TSFI of the base component, the**
17095 **subsystems associated with the interface will be subsystems considered in the Objectives**

17096 **17.4.3.4** The objectives of this sub-activity are to determine whether the formal security policy model of
17097 the TSF clearly and consistently describes the rules and characteristics of the security policies
17098 and whether this description corresponds with the description of security functions in the
17099 functional specification.

17100 **17.4.3.4 Input**

17101 **17.4.3.4** The evaluation evidence for this sub-activity is:

17102 **17.4.3.4** the ST;

17103 **17.4.3.4** the functional specification;

- 17104 **17.4.3.4** formal security policy model (ADV_SPM.1.1D);
- 17105 **17.4.3.4** formal proof of correspondence between the model and any formal functional
17106 specification (ADV_SPM.1.3D);
- 17107 **17.4.3.4** demonstration of correspondence between the model and the functional specification
17108 (ADV_SPM.1.4D).
- 17109 **17.4.3.4 Application notes**
- 17110 **17.4.3.4** This activity applies to cases where the developer has provided a formal security policy
17111 model of the TOE.
- 17112 **17.4.3.4** A formal TOE security policy model is a representation of the rules (synonymously
17113 termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
17114 terms. Their formal counterparts are called security properties and security features,
17115 respectively. The representation includes but is not limited to algebraic specifications, finite state
17116 machines and logic formalisms strong enough to formally infer the properties from the features.
17117 The formal TSP model is accompanied by an informal interpretation explaining how the rules and
17118 characteristics are mapped to the respective properties and features.
- 17119 **17.4.3.4** The creation of a formal security policy model helps to identify and eliminate
17120 ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
17121 has been built, the formal model serves the evaluation effort by contributing to the evaluator's
17122 judgement of how well the developer has understood the security functionality being
17123 implemented and whether there are inconsistencies between the security requirements and the
17124 TOE design. The confidence in the model is accompanied by a proof that it contains no
17125 inconsistencies.
- 17126 **17.4.3.4** A formal security model is a precise formal presentation of the important aspects of
17127 security and their relationship to the behaviour of the TOE; it identifies the set of rules
17128 (principles) that defines the TOE security policy and the set of practises (characteristics) that
17129 regulates how the TSF manages, protects, and otherwise controls the system resources. The
17130 model includes the set of restrictions and properties that specify how information and computing
17131 resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
17132 engineering arguments showing that these restrictions and properties play a key role in the
17133 enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
17134 as well as ancillary text to explain the model and to provide it with context. The security
17135 behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
17136 with the rest of the TOE and with its operational environment), as well as its internal behaviour.
- 17137 **17.4.3.4** The Security Policy Model of the TOE is informally abstracted from its realisation by
17138 considering the proposed security requirements of the ST. The informal abstraction is taken to be
17139 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
17140 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
17141 are always prone to fallacies; especially if relationships among subjects, objects and operations
17142 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
17143 characteristics of the security policy model are mapped to respective properties and features
17144 within some formal system, whose rigour and strength can afterwards be used to obtain the
17145 security properties by means of theorems and formal proof.
- 17146 **17.4.3.4** While the term “formal security policy model” is used in academic circles, the CC's
17147 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
17148 claimed. Therefore, the formal security policy model is merely a formal representation of the set
17149 of SFRs being claimed.
- 17150 **17.4.3.4** The term security policy has traditionally been associated with only access control
17151 policies, whether label-based (mandatory access control) or user-based (discretionary access

17152	control). However, a security policy is not limited to access control; there are also audit policies,
17153	identification policies, authentication policies, encryption policies, management policies, and any
17154	other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
17155	contains an assignment for identifying these policies that are formally modelled.
17156	17.4.3.4 It is recognized that not all policies can be formally modelled for all TOEs. This is
17157	because either a given policy can not be formally modelled in the otherwise well suited
17158	framework, or because the nature of the TOE renders impossible the modelling of policies that
17159	would otherwise be possible to model.
17160	17.4.3.4 Action ADV_SPM.1.1E
17161	17.4.3.4 ADV_SPM.1.1C <i>The model shall be in a formal style, supported by explanatory</i>
17162	<i>text as required, and identify the security policies of the TSF that are modelled.</i>
17163	17.4.3.4 Work unit ADV_SPM.1-1
17164	17.4.3.4 The evaluator <i>shall examine the TOE security policy model to determine that it is</i>
17165	written in a formal style.
17166	17.4.3.4 The evaluator identifies the formal framework upon which the TOE security policy
17167	model is based and ensures that it is founded on well established mathematical concepts. They
17168	also identify the security properties and features addressed in the application notes and ensure
17169	the formalization of at least one security policy.
17170	17.4.3.4 For guidance on formal methods refer to ISO/IEC 15408-3
17171	17.4.3.4 Work unit ADV_SPM.1-2
17172	17.4.3.4 The evaluator <i>shall examine the TOE security policy model to determine that it</i>
17173	contains all necessary informal explanatory text.
17174	17.4.3.4 Supporting narrative descriptions are necessary for all parts of the model (for example,
17175	to make clear the meaning of any formal notation and how they are used) including the security
17176	properties and features.
17177	17.4.3.4 Work unit ADV_SPM.1-3
17178	17.4.3.4 The evaluator <i>shall examine the TOE security policy model to determine that all</i>
17179	security policies of the TSF are identified that are modelled.
17180	17.4.3.4 The evaluator determines whether the SPM identifies the security policies for which a
17181	model is provided, identifying the relevant portions of the statement of SFRs that comprise each
17182	of the modelled policies.
17183	17.4.3.4 The evaluator determines whether the list of security policies identified by the SPM is
17184	consistent with the assignment of ADV_SPM.1.1D in the ST.
17185	17.4.3.4 The evaluator determines whether for each security policy identified by the SPM a
17186	model is in fact provided.
17187	17.4.3.4 ADV_SPM.1.2C <i>For all policies that are modelled, the model shall define</i>
17188	<i>security for the TOE and provide a formal proof that the TOE cannot reach a state that is</i>
17189	<i>not secure.</i>

- 17190 **17.4.3.4 Work unit ADV_SPM.1-4**
- 17191 **17.4.3.4** The evaluator *shall examine the principles and characteristics of the security*
 17192 policies to determine that the modelled security behaviour of the TOE is clearly articulated.
- 17193 **17.4.3.4** The security policies are expressed in terms of security principles (rules) which are
 17194 modelled by security properties and define the secure state of the TOE. For example, a model
 17195 based on state transitions could describe the security policies in terms of principles of its states,
 17196 identify its initial state, and define what it means to be a secure state.
- 17197 **17.4.3.4** The evaluator determines that the security policies are reflected within their formal
 17198 counterparts of the TSP model.
- 17199 **17.4.3.4** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 17200 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 17201 resources including attributes and conditions of the TOE) which are modelled by security
 17202 features. For example, a model based on state transitions could describe the characteristics as
 17203 possible actions in each secure state in a level of detail sufficient to decide into which state the
 17204 TOE will be transformed by that action.
- 17205 **17.4.3.4** Together the security principles and characteristics describe the entire security posture
 17206 of the TOE.
- 17207 **17.4.3.4** In the context of a formal TOE security policy model the security behaviour is
 17208 considered to be clearly articulated only if an adequate mapping from principles and
 17209 characteristics to their respective formal counterparts properties and features has been given.
 17210 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 17211 detailed enough to allow for correct identification of all security objectives and the relation to the
 17212 security environment.
- 17213 **17.4.3.4** The above condition for clear articulation is necessary but not sufficient. An informal
 17214 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 17215 must be provided in order to make clear their intended meaning.
- 17216 **17.4.3.4 Work unit ADV_SPM.1-5**
- 17217 **17.4.3.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
 17218 it formally proves that the security features enforce the security properties.
- 17219 **17.4.3.4** To determine the enforcement, the evaluator considers the security properties and the
 17220 security features and verifies that the arguments used in the proof are valid. The proof of
 17221 correspondence between the security properties and the security features shall be formal.
- 17222 **17.4.3.4** The validity of the security properties shall mean that the TOE is in a secure state. By
 17223 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 17224 state.
- 17225 **17.4.3.4 Work unit ADV_SPM.1-6**
- 17226 **17.4.3.4** The evaluator *shall examine the TOE security policy model rationale to determine that*
 17227 it proves the internal consistency of the TOE security policy model.
- 17228 **17.4.3.4** The proof shall show the absence of contradictions within the TOE security policy
 17229 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 17230 used in the proof are valid.
- 17231 **17.4.3.4** Since the TOE security policy model is formal, the proof of its internal consistency shall
 17232 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE

17233	security policy model usually is not possible due to the fundamental nature of formal frameworks.
17234	Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
17235	security policy model that prove the internal consistency by means of a combination with generic
17236	arguments of the formal framework.
17237	17.4.3.4 ADV_SPM.1.3C <i>The correspondence between the model and the functional</i>
17238	<i>specification shall be at the correct level of formality.</i>
17239	17.4.3.4 Work unit ADV_SPM.1-7
17240	17.4.3.4 The evaluator <i>shall examine the correspondence between the model and the functional</i>
17241	specification to determine that a semiformal demonstration of correspondence between the
17242	model and any semiformal functional specification is provided.
17243	17.4.3.4 This work unit is only applicable to a semiformal presentation of the functional
17244	specification, which is required by ADV_FSP.5.2C.
17245	17.4.3.4 A semiformal correspondence is one that results from a structured approach with a
17246	substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
17247	mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
17248	terms, and so it provides less ambiguity than would exist in an informal correspondence.
17249	17.4.3.4 For guidance on semiformal methods refer to Annex 3.1.1 ‘Semiformal and formal
17250	methods’.
17251	17.4.3.4 Work unit ADV_SPM.1-8
17252	17.4.3.4 The evaluator <i>shall examine the correspondence between the model and the functional</i>
17253	specification to determine that a formal proof of correspondence between the model and any
17254	formal functional specification is provided.
17255	17.4.3.4 This work unit is only applicable to a formal presentation of the functional specification,
17256	which is required by ADV_FSP.6.2D.
17257	17.4.3.4 There should be a formal proof of correspondence between the model and any formal
17258	functional specification.
17259	17.4.3.4 The formal proof of correspondence removes all subjective interpretations of its terms
17260	by enlisting well-established mathematical concepts to define the syntax and semantics of the
17261	formal notation and uses rules that support logical reasoning. The security features within the
17262	TOE (which are identified in the formal TSP model) are expressed in a formal specification
17263	language and shown to be satisfied by the formal specification.
17264	17.4.3.4 For guidance on formal methods refer to ISO/IEC 15408-3 .
17265	17.4.3.4 ADV_SPM.1.4C <i>The correspondence shall show that the functional</i>
17266	<i>specification is consistent and complete with respect to the model.</i>
17267	17.4.3.4 Work unit ADV_SPM.1-9
17268	17.4.3.4 The evaluator <i>shall examine the correspondence to determine that the behaviour at the</i>
17269	TSF interfaces (as articulated in the functional specification) is complete with respect to the
17270	behaviour modelled by the security features.
17271	17.4.3.4 The term “correspondence” here means both the formal proof of correspondence
17272	between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
17273	of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.

- 17274 **17.4.3.4** In determining completeness of the correspondence, the evaluator considers the
 17275 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 17276 features of the TSP model. The demonstration should show that all characteristics belonging to
 17277 policies that are required to be modelled have an associated feature description in the TOE
 17278 security policy model, and that each feature of the TSP model does occur in the mapping.
- 17279 **17.4.3.4** Abstention from formally modelling TSFI behaviour always calls for justification on the
 17280 developer's side (also confer the application notes above).
- 17281 **17.4.3.4 Work unit ADV_SPM.1-10**
- 17282 **17.4.3.4** The evaluator *shall examine the correspondence to determine that the behaviour at the*
 17283 TSF interfaces (as articulated in the functional specification) is consistent with respect to the
 17284 behaviour modelled by the security features.
- 17285 **17.4.3.4** The term "correspondence" here means both the formal proof of correspondence
 17286 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 17287 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 17288 **17.4.3.4** The meaning of consistency reflects the conventional understanding in contrast to the
 17289 internal consistency concept of work unit ADV_SPM.1-6.
- 17290 **17.4.3.4** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 17291 security features established in the preceding work unit and verifies that the correspondence
 17292 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 17293 behaviour.
- 17294 **17.4.3.4** For example, if TSFI behaviour dealt with access management on the granularity of
 17295 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 17296 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 17297 management for groups of users, then a TSP model describing the security behaviour of the TOE
 17298 in terms of individual users would also not be consistent.
- 17299 **17.4.3.4** As another example, if remote untrusted users had to pass more stringent
 17300 authentication procedures than administrators whose only point of access were within a
 17301 physically-protected area, then this difference in authentication procedures had to be reflected in
 17302 the security features.
- 17303 **17.4.3.4** TOE design (ADV_TDS) activity during the base component evaluation. The interfaces
 17304 on which the dependent component relies that did not form part of the TSFI of the base component
 17305 will map to subsystems outside of the base component TSF.
- 17306 ISO/IEC 15408-3 ACO_DEV.3.3C: *The development information shall provide a high-level description*
 17307 *of the behaviour of the base component subsystems, which support the enforcement of the dependent*
 17308 *component SFRs.*
- 17309 **17.4.3.7.11 Work unit ACO_DEV.3-4**
- 17310 The evaluator *shall examine* the development information to determine that it describes the
 17311 behaviour of the base component subsystems that support the enforcement of the dependent
 17312 component SFRs.
- 17313 The dependent component invokes interfaces of the base component for the provision of services
 17314 by the base component. For the interfaces of the base component that are invoked, the
 17315 development information shall provide a high-level description of the associated security
 17316 behaviour of the base component. The description of the base component security behaviour will
 17317 outline how the base component provides the necessary service when the call to the interface is
 17318 made. This description is to be at a level similar to that provided for ADV_TDS.1.4C. Therefore, the

17319 provision of the TOE design evidence from the base component evaluation would satisfy this work
17320 unit, where the interfaces invoked by the dependent component are TSFI of the base component. If
17321 the interfaces invoked by the dependent component are not TSFIs of the base component it is the
17322 associated security behaviour will not necessarily be described in the base component TOE design
17323 evidence.

17324 ISO/IEC 15408-3 ACO_DEV.3.4C: *The development information shall provide a mapping from the*
17325 *interfaces to the subsystems of the base component.*

17326 **17.4.3.7.12 Work unit ACO_DEV.3-5**

17327 The evaluator ***shall examine*** the development information to determine that the correspondence
17328 between the interfaces and subsystems of the base component is accurate.

17329 If the TOE design and functional specification evidence from the base component evaluation is
17330 available, this can be used to verify the accuracy of the correspondence between the interfaces and
17331 subsystems of the base component as used in the composed TOE. Those interfaces of the base
17332 component, which formed part of the base component TSFI will be described in the base
17333 component functional specification, and the associated subsystems will be described in the base
17334 component TOE design evidence. The tracing between the two will be provided in the base
17335 component TOE design evidence.

17336 If, however, the base component interface did not form part of the TSFI of the base component, the
17337 description of the subsystem behaviour provided in the development information will be used to
17338 verify the accuracy of the correspondence.

17339 ISO/IEC 15408-3 ACO_DEV.3.5C: *The development information shall show correspondence between*
17340 *the interfaces, used in the composed TOE, of the base component and the dependent component to*
17341 *support the TSF of the dependent component.*

17342 **17.4.3.7.13 Work unit ACO_DEV.3-6**

17343 The evaluator ***shall examine*** the development information to determine the correspondence,
17344 between the interfaces of the base component and the interfaces on which the dependent
17345 component relies, is accurate.

17346 The correspondence between the interfaces of the base component and the interfaces on which the
17347 dependent component relies may take the form of a matrix or table. The interfaces that are relied
17348 upon by the dependent component are identified in the reliance information (as examined during
17349 Reliance of dependent component (ACO_REL)).

17350 There is, during this activity, no requirement to determine completeness of the coverage of
17351 interfaces that are relied upon by the dependent component, only that the correspondence is
17352 correct and ensuring that interfaces of the base component are mapped to interfaces required by
17353 the dependent component wherever possible. The completeness of the coverage is considered in
17354 Composition rationale (ACO_COR) activities.

17355 **17.4.3.8 Action ACO_DEV.3.2E**

17356 **17.4.3.8.1 Work unit ACO_DEV.3-7**

17357 The evaluator ***shall examine*** the development information and the reliance information to
17358 determine that the interfaces are described consistently.

17359 The evaluator's goal in this work unit is to determine that the interfaces described in the
17360 development information for the base component and the reliance information for the dependent
17361 component are represented consistently.

17362 **17.5 Reliance of dependent component (ACO_REL)**

17363 **17.5.1 Evaluation of sub-activity (ACO_REL.1)**

17364 **17.5.1.1 Objectives**

17365 The objectives of this sub-activity are to determine whether the developer's reliance evidence
17366 provides sufficient information to determine that the necessary functionality is available in the
17367 base component, and the means by which that functionality is invoked. These are provided in
17368 terms of a high-level description.

17369 **17.5.1.2 Input**

17370 The evaluation evidence for this sub-activity is:

- 17371 a) the composed ST;
- 17372 b) the dependent component functional specification;
- 17373 c) the dependent component design;
- 17374 d) the dependent component architectural design;
- 17375 e) the reliance information.

17376 **17.5.1.3 Application notes**

17377 A dependent component whose TSF interacts with the base component requires functionality
17378 provided by that base component (e.g., remote authentication, remote audit data storage). In these
17379 cases, those invoked services need to be described for those charged with configuring the
17380 composed TOE for end users. The rationale for requiring this documentation is to aid integrators of
17381 the composed TOE to determine what services in the base component might have adverse effects
17382 on the dependent component, and to provide information against which to determine the
17383 compatibility of the components when applying the Development evidence (ACO_DEV) family.

17384 **17.5.1.4 Action ACO_REL.1.1E**

17385 ISO/IEC 15408-3 ACO_REL.1.1C: *The reliance information shall describe the functionality of the base*
17386 *component hardware, firmware and/or software that is relied upon by the dependent component TSF.*

17387 **17.5.1.4.1 Work unit ACO_REL.1-1**

17388 The evaluator ***shall check*** the reliance information to determine that it describes the functionality
17389 of the base dependent hardware, firmware and/or software that is relied upon by the dependent
17390 component TSF.

17391 The evaluator assesses the description of the security functionality that the dependent component
17392 TSF requires to be provided by the base component's hardware, firmware and software. The
17393 emphasis of this work unit is on the level of detail of this description, rather than on an assessment
17394 of the information's accuracy. (The assessment of the accuracy of the information is the focus of the
17395 next work unit.)

17396 **17.5.1.5 This description of the base component's functionality need not be any more**
17397 **detailed than the level of the description of a component of the TSF, as would be provided**
17398 **in the TOE Design (Objectives)**

17399 **17.5.1.5** The objectives of this sub-activity are to determine whether the formal security policy model of
17400 the TSF clearly and consistently describes the rules and characteristics of the security policies

17401	and whether this description corresponds with the description of security functions in the
17402	functional specification.
17403	17.5.1.5 Input
17404	17.5.1.5 The evaluation evidence for this sub-activity is:
17405	17.5.1.5 the ST;
17406	17.5.1.5 the functional specification;
17407	17.5.1.5 formal security policy model (ADV_SPM.1.1D);
17408	17.5.1.5 formal proof of correspondence between the model and any formal functional
17409	specification (ADV_SPM.1.3D);
17410	17.5.1.5 demonstration of correspondence between the model and the functional specification
17411	(ADV_SPM.1.4D).
17412	17.5.1.5 Application notes
17413	17.5.1.5 This activity applies to cases where the developer has provided a formal security policy
17414	model of the TOE.
17415	17.5.1.5 A formal TOE security policy model is a representation of the rules (synonymously
17416	termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
17417	terms. Their formal counterparts are called security properties and security features,
17418	respectively. The representation includes but is not limited to algebraic specifications, finite state
17419	machines and logic formalisms strong enough to formally infer the properties from the features.
17420	The formal TSP model is accompanied by an informal interpretation explaining how the rules and
17421	characteristics are mapped to the respective properties and features.
17422	17.5.1.5 The creation of a formal security policy model helps to identify and eliminate
17423	ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
17424	has been built, the formal model serves the evaluation effort by contributing to the evaluator's
17425	judgement of how well the developer has understood the security functionality being
17426	implemented and whether there are inconsistencies between the security requirements and the
17427	TOE design. The confidence in the model is accompanied by a proof that it contains no
17428	inconsistencies.
17429	17.5.1.5 A formal security model is a precise formal presentation of the important aspects of
17430	security and their relationship to the behaviour of the TOE; it identifies the set of rules
17431	(principles) that defines the TOE security policy and the set of practises (characteristics) that
17432	regulates how the TSF manages, protects, and otherwise controls the system resources. The
17433	model includes the set of restrictions and properties that specify how information and computing
17434	resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
17435	engineering arguments showing that these restrictions and properties play a key role in the
17436	enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
17437	as well as ancillary text to explain the model and to provide it with context. The security
17438	behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
17439	with the rest of the TOE and with its operational environment), as well as its internal behaviour.
17440	17.5.1.5 The Security Policy Model of the TOE is informally abstracted from its realisation by
17441	considering the proposed security requirements of the ST. The informal abstraction is taken to be
17442	successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
17443	formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
17444	are always prone to fallacies; especially if relationships among subjects, objects and operations
17445	get more and more involved. In order to minimise the risk of insecure state arrivals the rules and

- 17446 characteristics of the security policy model are mapped to respective properties and features
 17447 within some formal system, whose rigour and strength can afterwards be used to obtain the
 17448 security properties by means of theorems and formal proof.
- 17449 **17.5.1.5** While the term “formal security policy model” is used in academic circles, the CC's
 17450 approach has no fixed definition of “security”; it would equate to whatever SFRs are being
 17451 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 17452 of SFRs being claimed.
- 17453 **17.5.1.5** The term security policy has traditionally been associated with only access control
 17454 policies, whether label-based (mandatory access control) or user-based (discretionary access
 17455 control). However, a security policy is not limited to access control; there are also audit policies,
 17456 identification policies, authentication policies, encryption policies, management policies, and any
 17457 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 17458 contains an assignment for identifying these policies that are formally modelled.
- 17459 **17.5.1.5** It is recognized that not all policies can be formally modelled for all TOEs. This is
 17460 because either a given policy can not be formally modelled in the otherwise well suited
 17461 framework, or because the nature of the TOE renders impossible the modelling of policies that
 17462 would otherwise be possible to model.
- 17463 **17.5.1.5 Action ADV_SPM.1.1E**
- 17464 **17.5.1.5 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 17465 *text as required, and identify the security policies of the TSF that are modelled.*
- 17466 **17.5.1.5 Work unit ADV_SPM.1-1**
- 17467 **17.5.1.5** The evaluator *shall examine the TOE security policy model to determine that it is*
 17468 *written in a formal style.*
- 17469 **17.5.1.5** The evaluator identifies the formal framework upon which the TOE security policy
 17470 model is based and ensures that it is founded on well established mathematical concepts. **They**
 17471 **also identify the security properties and features addressed in the application notes and ensure**
 17472 **the formalization of at least one security policy.**
- 17473 **17.5.1.5** For guidance on formal methods refer to ISO/IEC 15408-3
- 17474 **17.5.1.5 Work unit ADV_SPM.1-2**
- 17475 **17.5.1.5** The evaluator *shall examine the TOE security policy model to determine that it*
 17476 *contains all necessary informal explanatory text.*
- 17477 **17.5.1.5** Supporting narrative descriptions are necessary for all parts of the model (for example,
 17478 to make clear the meaning of any formal notation and how they are used) including the security
 17479 properties and features.
- 17480 **17.5.1.5 Work unit ADV_SPM.1-3**
- 17481 **17.5.1.5** The evaluator *shall examine the TOE security policy model to determine that all*
 17482 *security policies of the TSF are identified that are modelled.*
- 17483 **17.5.1.5** The evaluator determines whether the SPM identifies the security policies for which a
 17484 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 17485 of the modelled policies.
- 17486 **17.5.1.5** The evaluator determines whether the list of security policies identified by the SPM is
 17487 consistent with the assignment of ADV_SPM.1.1D in the ST.

17488	17.5.1.5 The evaluator determines whether for each security policy identified by the SPM a
17489	model is in fact provided.
17490	17.5.1.5 ADV_SPM.1.2C <i>For all policies that are modelled, the model shall define</i>
17491	<i>security for the TOE and provide a formal proof that the TOE cannot reach a state that is</i>
17492	<i>not secure.</i>
17493	17.5.1.5 Work unit ADV_SPM.1-4
17494	17.5.1.5 The evaluator <i>shall examine the principles and characteristics of the security policies</i>
17495	to determine that the modelled security behaviour of the TOE is clearly articulated.
17496	17.5.1.5 The security policies are expressed in terms of security principles (rules) which are
17497	modelled by security properties and define the secure state of the TOE. For example, a model
17498	based on state transitions could describe the security policies in terms of principles of its states,
17499	identify its initial state, and define what it means to be a secure state.
17500	17.5.1.5 The evaluator determines that the security policies are reflected within their formal
17501	counterparts of the TSP model.
17502	17.5.1.5 The TOE security behaviour is expressed in terms of security characteristics (i.e.
17503	portions of TOE security functionality managing, protecting, and otherwise controlling the system
17504	resources including attributes and conditions of the TOE) which are modelled by security
17505	features. For example, a model based on state transitions could describe the characteristics as
17506	possible actions in each secure state in a level of detail sufficient to decide into which state the
17507	TOE will be transformed by that action.
17508	17.5.1.5 Together the security principles and characteristics describe the entire security posture
17509	of the TOE.
17510	17.5.1.5 In the context of a formal TOE security policy model the security behaviour is
17511	considered to be clearly articulated only if an adequate mapping from principles and
17512	characteristics to their respective formal counterparts properties and features has been given.
17513	The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
17514	detailed enough to allow for correct identification of all security objectives and the relation to the
17515	security environment.
17516	17.5.1.5 The above condition for clear articulation is necessary but not sufficient. An informal
17517	interpretation of all formal concepts (including attributes, predicates and variables, if available)
17518	must be provided in order to make clear their intended meaning.
17519	17.5.1.5 Work unit ADV_SPM.1-5
17520	17.5.1.5 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
17521	it formally proves that the security features enforce the security properties.
17522	17.5.1.5 To determine the enforcement, the evaluator considers the security properties and the
17523	security features and verifies that the arguments used in the proof are valid. The proof of
17524	correspondence between the security properties and the security features shall be formal.
17525	17.5.1.5 The validity of the security properties shall mean that the TOE is in a secure state. By
17526	this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
17527	state.
17528	17.5.1.5 Work unit ADV_SPM.1-6
17529	17.5.1.5 The evaluator <i>shall examine the TOE security policy model rationale to determine that</i>
17530	it proves the internal consistency of the TOE security policy model.

- 17531 **17.5.1.5** The proof shall show the absence of contradictions within the TOE security policy
 17532 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 17533 used in the proof are valid.
- 17534 **17.5.1.5** Since the TOE security policy model is formal, the proof of its internal consistency shall
 17535 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 17536 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 17537 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 17538 security policy model that prove the internal consistency by means of a combination with generic
 17539 arguments of the formal framework.
- 17540 **17.5.1.5 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 17541 *specification shall be at the correct level of formality.*
- 17542 **17.5.1.5 Work unit ADV_SPM.1-7**
- 17543 **17.5.1.5** The evaluator *shall examine the correspondence between the model and the functional*
 17544 *specification* to determine that a semiformal demonstration of correspondence between the
 17545 model and any semiformal functional specification is provided.
- 17546 **17.5.1.5** This work unit is only applicable to a semiformal presentation of the functional
 17547 specification, which is required by ADV_FSP.5.2C.
- 17548 **17.5.1.5** A semiformal correspondence is one that results from a structured approach with a
 17549 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 17550 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 17551 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 17552 **17.5.1.5** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 17553 **methods**'.
- 17554 **17.5.1.5 Work unit ADV_SPM.1-8**
- 17555 **17.5.1.5** The evaluator *shall examine the correspondence between the model and the functional*
 17556 *specification* to determine that a formal proof of correspondence between the model and any
 17557 formal functional specification is provided.
- 17558 **17.5.1.5** This work unit is only applicable to a formal presentation of the functional specification,
 17559 which is required by ADV_FSP.6.2D.
- 17560 **17.5.1.5** There should be a formal proof of correspondence between the model and any formal
 17561 functional specification.
- 17562 **17.5.1.5** The formal proof of correspondence removes all subjective interpretations of its terms
 17563 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 17564 formal notation and uses rules that support logical reasoning. The security features within the
 17565 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 17566 language and shown to be satisfied by the formal specification.
- 17567 **17.5.1.5** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 17568 **17.5.1.5 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 17569 *specification is consistent and complete with respect to the model.*

17570	17.5.1.5 Work unit ADV_SPM.1-9
17571	17.5.1.5 The evaluator <i>shall examine the correspondence to determine that the behaviour</i>
17572	
17573	
17574	17.5.1.5 The term “correspondence” here means both the formal proof of correspondence
17575	
17576	
17577	17.5.1.5 In determining completeness of the correspondence, the evaluator considers the
17578	
17579	
17580	
17581	
17582	17.5.1.5 Abstention from formally modelling TSFI behaviour always calls for justification on the
17583	
17584	17.5.1.5 Work unit ADV_SPM.1-10
17585	17.5.1.5 The evaluator <i>shall examine the correspondence to determine that the behaviour at the</i>
17586	
17587	
17588	17.5.1.5 The term “correspondence” here means both the formal proof of correspondence
17589	
17590	
17591	17.5.1.5 The meaning of consistency reflects the conventional understanding in contrast to the
17592	
17593	17.5.1.5 In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
17594	
17595	
17596	
17597	17.5.1.5 For example, if TSFI behaviour dealt with access management on the granularity of
17598	
17599	
17600	
17601	
17602	17.5.1.5 As another example, if remote untrusted users had to pass more stringent
17603	
17604	
17605	
17606	17.5.1.5 TOE design (ADV_TDS))
17607	17.5.1.8.11 Work unit ACO_REL.1-2
17608	The evaluator <i>shall examine</i> the reliance information to determine that it accurately reflects the
17609	
17610	The reliance information contains the description of the base component's security functionality
17611	
17612	
17613	

- 17613 compares the reliance information with the statement of objectives for the environment in the ST
17614 for the dependent component.
- 17615 For example, if the reliance information claims that the dependent component TSF relies upon the
17616 base component to store and protect audit data, yet other evaluation evidence (e.g. the dependent
17617 component design) makes it clear that the dependent component TSF itself is storing and
17618 protecting the audit data, this would indicate an inaccuracy.
- 17619 It should be noted that the objectives for the operational environment may include objectives that
17620 can be met by non-IT measures. While the services that the base component environment is
17621 expected to provide may be described in the description of IT objectives for the operational
17622 environment in the dependent component ST, it is not required that all such expectations on the
17623 environment be described in the reliance information.
- 17624 ISO/IEC 15408-3 ACO_REL.1.2C: *The reliance information shall describe all interactions through*
17625 *which the dependent component TSF requests services from the base component.*
- 17626 **17.5.1.8.12 Work unit ACO_REL.1-3**
- 17627 The evaluator ***shall examine*** the reliance information to determine that it describes all
17628 interactions between the dependent component and the base component, through which the
17629 dependent component TSF requests services from the base component.
- 17630 The dependent component TSF may request services of the base component that were not within
17631 the TSF of the base component (see **B.3, Interactions between composed IT entities** in ISO/IEC
17632 15408-3).
- 17633 The interfaces to the base component's functionality are described at the same level as the
17634 description of the interfaces to the dependent component TSF functionality, as would be provided
17635 between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)).
- 17636 The purpose of describing the interactions between the dependent component and the base
17637 component is to provide an understanding of how the dependent component TSF relies upon the
17638 base component for the provision of services to support the operation of security functionality of
17639 the dependent component. These interactions do not need to be characterised at the
17640 implementation level (e.g. parameters passed from one routine in a component to a routine in
17641 another component), but the data elements identified for a particular component that are going to
17642 be used by another component should be covered in this description. The statement should help
17643 the reader understand in general why the interaction is necessary.
- 17644 Accuracy and completeness of the interfaces is based on the security functionality that the TSF
17645 requires to be provided by the base component, as assessed in work units ACO_REL.1-1 and
17646 ACO_REL.1-2. It should be possible to map all of the functionality described in the earlier work
17647 units to the interfaces identified in this work unit, and vice versa. An interface that does not
17648 correspond to described functionality would also indicate an inadequacy.
- 17649 ISO/IEC 15408-3 ACO_REL.1.3C: *The reliance information shall describe how the dependent TSF*
17650 *protects itself from interference and tampering by the base component.*
- 17651 **17.5.1.8.13 Work unit ACO_REL.1-4**
- 17652 The evaluator ***shall examine*** the reliance information to determine that it describes how the
17653 dependent TSF protects itself from interference and tampering by the base component.
- 17654 The description of how the dependent component protects itself from interference and tampering
17655 by the base component is to be provided at the same level of detail as necessary for ADV_ARC.1-4.

17656 **17.5.2 Evaluation of sub-activity (ACO_REL.2)**

17657 **17.5.2.1 Objectives**

17658 The objectives of this sub-activity are to determine whether the developer's reliance evidence
17659 provides sufficient information to determine that the necessary functionality is available in the
17660 base component, and the means by which that functionality is invoked. This is provided in terms of
17661 the interfaces between the dependent and base component and the return values from those
17662 interfaces called by the dependent component.

17663 **17.5.2.2 Input**

17664 The evaluation evidence for this sub-activity is:

- 17665 a) the composed ST;
- 17666 b) the dependent component functional specification;
- 17667 c) the dependent component design;
- 17668 d) the dependent component implementation representation;
- 17669 e) the dependent component architectural design;
- 17670 f) the reliance information.

17671 **17.5.2.3 Application notes**

17672 A dependent component whose TSF interacts with the base component requires functionality
17673 provided by that base component (e.g., remote authentication, remote audit data storage). In these
17674 cases, those invoked services need to be described for those charged with configuring the
17675 composed TOE for end users. The rationale for requiring this documentation is to aid integrators of
17676 the composed TOE to determine what services in the base component might have adverse effects
17677 on the dependent component, and to provide information against which to determine the
17678 compatibility of the components when applying the Development evidence (ACO_DEV) family.

17679 **17.5.2.4 Action ACO_REL.2.1E**

17680 ISO/IEC 15408-3 ACO_REL.2.1C: *The reliance information shall describe the functionality of the base*
17681 *component hardware, firmware and/or software that is relied upon by the dependent component TSF.*

17682 **17.5.2.4.1 Work unit ACO_REL.2-1**

17683 The evaluator ***shall check*** the reliance information to determine that it describes the functionality
17684 of the base dependent hardware, firmware and/or software that is relied upon by the dependent
17685 component TSF.

17686 The evaluator assesses the description of the security functionality that the dependent component
17687 TSF requires to be provided by the base component's hardware, firmware and software. The
17688 emphasis of this work unit is on the level of detail of this description, rather than on an assessment
17689 of the information's accuracy. (The assessment of the accuracy of the information is the focus of the
17690 next work unit.)

17691	17.5.2.5 This description of the base component's functionality need not be any more
17692	detailed than the level of the description of a component of the TSF, as would be provided
17693	in the TOE Design (Objectives)
17694	17.5.2.5 The objectives of this sub-activity are to determine whether the formal security policy model of
17695	the TSF clearly and consistently describes the rules and characteristics of the security policies
17696	and whether this description corresponds with the description of security functions in the
17697	functional specification.
17698	17.5.2.5 Input
17699	17.5.2.5 The evaluation evidence for this sub-activity is:
17700	17.5.2.5 the ST;
17701	17.5.2.5 the functional specification;
17702	17.5.2.5 formal security policy model (ADV_SPM.1.1D);
17703	17.5.2.5 formal proof of correspondence between the model and any formal functional specification
17704	(ADV_SPM.1.3D);
17705	17.5.2.5 demonstration of correspondence between the model and the functional specification
17706	(ADV_SPM.1.4D).
17707	17.5.2.5 Application notes
17708	17.5.2.5 This activity applies to cases where the developer has provided a formal security policy model of
17709	the TOE.
17710	17.5.2.5 A formal TOE security policy model is a representation of the rules (synonymously termed
17711	"principles") of security policies and characteristics of the TSF behaviour in mathematical terms.
17712	Their formal counterparts are called security properties and security features, respectively. The
17713	representation includes but is not limited to algebraic specifications, finite state machines and
17714	logic formalisms strong enough to formally infer the properties from the features. The formal TSP
17715	model is accompanied by an informal interpretation explaining how the rules and characteristics
17716	are mapped to the respective properties and features.
17717	17.5.2.5 The creation of a formal security policy model helps to identify and eliminate ambiguous,
17718	inconsistent, contradictory, or unenforceable security policy elements. Once the TOE has been
17719	built, the formal model serves the evaluation effort by contributing to the evaluator's judgement
17720	of how well the developer has understood the security functionality being implemented and
17721	whether there are inconsistencies between the security requirements and the TOE design. The
17722	confidence in the model is accompanied by a proof that it contains no inconsistencies.
17723	17.5.2.5 A formal security model is a precise formal presentation of the important aspects of security and
17724	their relationship to the behaviour of the TOE; it identifies the set of rules (principles) that
17725	defines the TOE security policy and the set of practises (characteristics) that regulates how the
17726	TSF manages, protects, and otherwise controls the system resources. The model includes the set
17727	of restrictions and properties that specify how information and computing resources are
17728	prevented from being used to violate the SFRs, accompanied by a persuasive set of engineering
17729	arguments showing that these restrictions and properties play a key role in the enforcement of
17730	the SFRs. It consists both of the formalisms that express the security functionality, as well as
17731	ancillary text to explain the model and to provide it with context. The security behaviour of the
17732	TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts with the rest of
17733	the TOE and with its operational environment), as well as its internal behaviour.

- 17734 **17.5.2.5** The Security Policy Model of the TOE is informally abstracted from its realisation by
 17735 considering the proposed security requirements of the ST. The informal abstraction is taken to be
 17736 successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
 17737 formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
 17738 are always prone to fallacies; especially if relationships among subjects, objects and operations
 17739 get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
 17740 characteristics of the security policy model are mapped to respective properties and features
 17741 within some formal system, whose rigour and strength can afterwards be used to obtain the
 17742 security properties by means of theorems and formal proof.
- 17743 **17.5.2.5** While the term "formal security policy model" is used in academic circles, the CC's
 17744 approach has no fixed definition of "security"; it would equate to whatever SFRs are being
 17745 claimed. Therefore, the formal security policy model is merely a formal representation of the set
 17746 of SFRs being claimed.
- 17747 **17.5.2.5** The term security policy has traditionally been associated with only access control
 17748 policies, whether label-based (mandatory access control) or user-based (discretionary access
 17749 control). However, a security policy is not limited to access control; there are also audit policies,
 17750 identification policies, authentication policies, encryption policies, management policies, and any
 17751 other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
 17752 contains an assignment for identifying these policies that are formally modelled.
- 17753 **17.5.2.5** It is recognized that not all policies can be formally modelled for all TOEs. This is
 17754 because either a given policy can not be formally modelled in the otherwise well suited
 17755 framework, or because the nature of the TOE renders impossible the modelling of policies that
 17756 would otherwise be possible to model.
- 17757 **17.5.2.5 Action ADV_SPM.1.1E**
- 17758 **17.5.2.5 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 17759 *text as required, and identify the security policies of the TSF that are modelled.*
- 17760 **17.5.2.5 Work unit ADV_SPM.1-1**
- 17761 **17.5.2.5** The evaluator ***shall examine the TOE security policy model to determine that it is***
 17762 *written in a formal style.*
- 17763 **17.5.2.5** The evaluator identifies the formal framework upon which the TOE security policy
 17764 model is based and ensures that it is founded on well established mathematical concepts. **They**
 17765 **also identify the security properties and features addressed in the application notes and ensure**
 17766 **the formalization of at least one security policy.**
- 17767 **17.5.2.5** For guidance on formal methods refer to ISO/IEC 15408-3
- 17768 **17.5.2.5 Work unit ADV_SPM.1-2**
- 17769 **17.5.2.5** The evaluator ***shall examine the TOE security policy model to determine that it***
 17770 *contains all necessary informal explanatory text.*
- 17771 **17.5.2.5** Supporting narrative descriptions are necessary for all parts of the model (for example,
 17772 to make clear the meaning of any formal notation and how they are used) including the security
 17773 properties and features.
- 17774 **17.5.2.5 Work unit ADV_SPM.1-3**
- 17775 **17.5.2.5** The evaluator ***shall examine the TOE security policy model to determine that all***
 17776 *security policies of the TSF are identified that are modelled.*

- 17777 **17.5.2.5** The evaluator determines whether the SPM identifies the security policies for which a
 17778 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 17779 of the modelled policies.
- 17780 **17.5.2.5** The evaluator determines whether the list of security policies identified by the SPM is
 17781 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 17782 **17.5.2.5** The evaluator determines whether for each security policy identified by the SPM a
 17783 model is in fact provided.
- 17784 **17.5.2.5 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 17785 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 17786 *not secure.*
- 17787 **17.5.2.5 Work unit ADV_SPM.1-4**
- 17788 **17.5.2.5** The evaluator *shall examine the principles and characteristics of the security policies*
 17789 *to determine that the modelled security behaviour of the TOE is clearly articulated.*
- 17790 **17.5.2.5** The security policies are expressed in terms of security principles (rules) which are
 17791 modelled by security properties and define the secure state of the TOE. For example, a model
 17792 based on state transitions could describe the security policies in terms of principles of its states,
 17793 identify its initial state, and define what it means to be a secure state.
- 17794 **17.5.2.5** The evaluator determines that the security policies are reflected within their formal
 17795 counterparts of the TSP model.
- 17796 **17.5.2.5** The TOE security behaviour is expressed in terms of security characteristics (i.e.
 17797 portions of TOE security functionality managing, protecting, and otherwise controlling the system
 17798 resources including attributes and conditions of the TOE) which are modelled by security
 17799 features. For example, a model based on state transitions could describe the characteristics as
 17800 possible actions in each secure state in a level of detail sufficient to decide into which state the
 17801 TOE will be transformed by that action.
- 17802 **17.5.2.5** Together the security principles and characteristics describe the entire security posture
 17803 of the TOE.
- 17804 **17.5.2.5** In the context of a formal TOE security policy model the security behaviour is
 17805 considered to be clearly articulated only if an adequate mapping from principles and
 17806 characteristics to their respective formal counterparts properties and features has been given.
 17807 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
 17808 detailed enough to allow for correct identification of all security objectives and the relation to the
 17809 security environment.
- 17810 **17.5.2.5** The above condition for clear articulation is necessary but not sufficient. An informal
 17811 interpretation of all formal concepts (including attributes, predicates and variables, if available)
 17812 must be provided in order to make clear their intended meaning.
- 17813 **17.5.2.5 Work unit ADV_SPM.1-5**
- 17814 **17.5.2.5** The evaluator *shall examine the TOE security policy model rationale to determine that*
 17815 *it formally proves that the security features enforce the security properties.*
- 17816 **17.5.2.5** To determine the enforcement, the evaluator considers the security properties and the
 17817 security features and verifies that the arguments used in the proof are valid. The proof of
 17818 correspondence between the security properties and the security features shall be formal.

- 17819 **17.5.2.5** The validity of the security properties shall mean that the TOE is in a secure state. By
 17820 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
 17821 state.
- 17822 **17.5.2.5 Work unit ADV_SPM.1-6**
- 17823 **17.5.2.5** The evaluator *shall examine the TOE security policy model rationale to determine that*
 17824 it proves the internal consistency of the TOE security policy model.
- 17825 **17.5.2.5** The proof shall show the absence of contradictions within the TOE security policy
 17826 model. In determining the absence of contradictions, the evaluator verifies that the arguments
 17827 used in the proof are valid.
- 17828 **17.5.2.5** Since the TOE security policy model is formal, the proof of its internal consistency shall
 17829 be formal. It is recognized that a complete formal proof of the internal consistency of the TOE
 17830 security policy model usually is not possible due to the fundamental nature of formal frameworks.
 17831 Generally, it is sufficient to generate evidence using formal proofs based on the specific TOE
 17832 security policy model that prove the internal consistency by means of a combination with generic
 17833 arguments of the formal framework.
- 17834 **17.5.2.5 ADV_SPM.1.3C** *The correspondence between the model and the functional*
 17835 *specification shall be at the correct level of formality.*
- 17836 **17.5.2.5 Work unit ADV_SPM.1-7**
- 17837 **17.5.2.5** The evaluator *shall examine the correspondence between the model and the functional*
 17838 *specification to determine that a semiformal demonstration of correspondence between the*
 17839 *model and any semiformal functional specification is provided.*
- 17840 **17.5.2.5** This work unit is only applicable to a semiformal presentation of the functional
 17841 specification, which is required by ADV_FSP.5.2C.
- 17842 **17.5.2.5** A semiformal correspondence is one that results from a structured approach with a
 17843 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 17844 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 17845 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 17846 **17.5.2.5** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 17847 **methods**'.
- 17848 **17.5.2.5 Work unit ADV_SPM.1-8**
- 17849 **17.5.2.5** The evaluator *shall examine the correspondence between the model and the functional*
 17850 *specification to determine that a formal proof of correspondence between the model and any*
 17851 *formal functional specification is provided.*
- 17852 **17.5.2.5** This work unit is only applicable to a formal presentation of the functional specification,
 17853 which is required by ADV_FSP.6.2D.
- 17854 **17.5.2.5** There should be a formal proof of correspondence between the model and any formal
 17855 functional specification.
- 17856 **17.5.2.5** The formal proof of correspondence removes all subjective interpretations of its terms
 17857 by enlisting well-established mathematical concepts to define the syntax and semantics of the
 17858 formal notation and uses rules that support logical reasoning. The security features within the
 17859 TOE (which are identified in the formal TSP model) are expressed in a formal specification
 17860 language and shown to be satisfied by the formal specification.

- 17861 **17.5.2.5** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 17862 **17.5.2.5 ADV_SPM.1.4C** *The correspondence shall show that the functional*
 17863 *specification is consistent and complete with respect to the model.*
- 17864 **17.5.2.5 Work unit ADV_SPM.1-9**
- 17865 **17.5.2.5** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 17866 ***TSF interfaces (as articulated in the functional specification) is complete with respect to the***
 17867 ***behaviour modelled by the security features.***
- 17868 **17.5.2.5** The term “correspondence” here means both the formal proof of correspondence
 17869 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 17870 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 17871 **17.5.2.5** In determining completeness of the correspondence, the evaluator considers the
 17872 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 17873 features of the TSP model. The demonstration should show that all characteristics belonging to
 17874 policies that are required to be modelled have an associated feature description in the TOE
 17875 security policy model, and that each feature of the TSP model does occur in the mapping.
- 17876 **17.5.2.5** Abstention from formally modelling TSFI behaviour always calls for justification on the
 17877 developer’s side (also confer the application notes above).
- 17878 **17.5.2.5 Work unit ADV_SPM.1-10**
- 17879 **17.5.2.5** The evaluator ***shall examine the correspondence to determine that the behaviour at the***
 17880 ***TSF interfaces (as articulated in the functional specification) is consistent with respect to the***
 17881 ***behaviour modelled by the security features.***
- 17882 **17.5.2.5** The term “correspondence” here means both the formal proof of correspondence
 17883 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 17884 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.
- 17885 **17.5.2.5** The meaning of consistency reflects the conventional understanding in contrast to the
 17886 internal consistency concept of work unit ADV_SPM.1-6.
- 17887 **17.5.2.5** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 17888 security features established in the preceding work unit and verifies that the correspondence
 17889 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 17890 behaviour.
- 17891 **17.5.2.5** For example, if TSFI behaviour dealt with access management on the granularity of
 17892 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 17893 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 17894 management for groups of users, then a TSP model describing the security behaviour of the TOE
 17895 in terms of individual users would also not be consistent.
- 17896 **17.5.2.5** As another example, if remote untrusted users had to pass more stringent
 17897 authentication procedures than administrators whose only point of access were within a
 17898 physically-protected area, then this difference in authentication procedures had to be reflected in
 17899 the security features.
- 17900 **17.5.2.5** TOE design (ADV_TDS))

17901 **17.5.2.8.11 Work unit ACO_REL.2-2**

17902 The evaluator ***shall examine*** the reliance information to determine that it accurately reflects the
17903 objectives specified for the operational environment of the dependent component.

17904 The reliance information contains the description of the base component's security functionality
17905 relied upon by the dependent component. To ensure that the reliance information is consistent
17906 with the expectations of the operational environment of the dependent component, the evaluator
17907 compares the reliance information with the statement of objectives for the environment in the ST
17908 for the dependent component.

17909 For example, if the reliance information claims that the dependent component TSF relies upon the
17910 base component to store and protect audit data, yet other evaluation evidence (e.g. the dependent
17911 component design) makes it clear that the dependent component TSF itself is storing and
17912 protecting the audit data, this would indicate an inaccuracy.

17913 It should be noted that the objectives for the operational environment may include objectives that
17914 can be met by non-IT measures. While the services that the base component environment is
17915 expected to provide may be described in the description of IT objectives for the operational
17916 environment in the dependent component ST, it is not required that all such expectations on the
17917 environment be described in the reliance information.

17918 ISO/IEC 15408-3 ACO_REL.2.2C: *The reliance information shall describe all interactions through*
17919 *which the dependent component TSF requests services from the base component.*

17920 **17.5.2.8.12 Work unit ACO_REL.2-3**

17921 The evaluator ***shall examine*** the reliance information to determine that it describes all
17922 interactions between the dependent component and the base component, through which the
17923 dependent component TSF requests services from the base component.

17924 The dependent component TSF may request services of the base component that were not within
17925 the TSF of the base component (see Annex **B.3, Interactions between composed IT entities** in
17926 ISO/IEC 15408-3).

17927 The interfaces to the base component's functionality are described at the same level as the
17928 description of the interfaces to the dependent component TSF functionality, as would be provided
17929 between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)).

17930 The purpose of describing the interactions between the dependent component and the base
17931 component is to provide an understanding of how the dependent component TSF relies upon the
17932 base component for the provision of services to support the operation of security functionality of
17933 the dependent component. These interactions do not need to be characterised at the
17934 implementation level (e.g. parameters passed from one routine in a component to a routine in
17935 another component), but the data elements identified for a particular component that are going to
17936 be used by another component should be covered in this description. The statement should help
17937 the reader understand in general why the interaction is necessary.

17938 Accuracy and completeness of the interfaces is based on the security functionality that the TSF
17939 requires to be provided by the base component, as assessed in work units ACO_REL.2-1 and
17940 ACO_REL.2-2. It should be possible to map all of the functionality described in the earlier work
17941 units to the interfaces identified in this work unit, and vice versa. An interface that does not
17942 correspond to described functionality would also indicate an inadequacy.

17943 ISO/IEC 15408-3 ACO_REL.2.3C: *The reliance information shall describe each interaction in terms of*
17944 *the interface used and the return values from those interfaces.*

17945 **17.5.2.8.13 Work unit ACO_REL.2-4**

17946 The reliance information shall describe each interaction in terms of the interface used and the
17947 return values from those interfaces.

17948 The identification of the interfaces used by the dependent component TSF when making services
17949 requests of the base component allows an integrator to determine whether the base component
17950 provides all the necessary corresponding interfaces. This understanding is further gained through
17951 the specification of the return values expected by the dependent component. The evaluator ensures
17952 that interfaces are described for each interaction specified (as analysed in ACO_REL.2-3).

17953 ISO/IEC 15408-3 ACO_REL.2.4C: *The reliance information shall describe how the dependent TSF*
17954 *protects itself from interference and tampering by the base component.*

17955 **17.5.2.8.14 Work unit ACO_REL.2-5**

17956 The evaluator **shall examine** the reliance information to determine that it describes how the
17957 dependent TSF protects itself from interference and tampering by the base component.

17958 The description of how the dependent component protects itself from interference and tampering
17959 by the base component is to be provided at the same level of detail as necessary for ADV_ARC.1-4.

17960 **17.6 Composed TOE testing (ACO_CTT)**17961 **17.6.1 Evaluation of sub-activity (ACO_CTT.1)**17962 **17.6.1.1 Objectives**

17963 The objective of this sub-activity is to determine whether the developer correctly performed and
17964 documented tests for each of the base component interfaces on which the dependent component
17965 relies. As part of this determination the evaluator repeats a sample of the tests performed by the
17966 developer and performs any additional tests required to ensure the expected behaviour of all
17967 composed TOE SFRs and interfaces of the base component relied upon by the dependent
17968 component is demonstrated.

17969 **17.6.1.2 Input**

17970 The evaluation evidence for this sub-activity is:

- 17971 a) the composed TOE suitable for testing;
- 17972 b) the composed TOE testing evidence;
- 17973 c) the reliance information;
- 17974 d) the development information.

17975 **17.6.1.3 Action ACO_CTT.1.1E**

17976 ISO/IEC 15408-3 ACO_CTT.1.1C: *The composed TOE and base component interface test*
17977 *documentation shall consist of test plans, expected test results and actual test results.*

17978 **17.6.1.3.1 Work unit ACO_CTT.1-1**

17979 The evaluator **shall examine** the composed TOE test documentation to determine that it consists
17980 of test plans, expected test results and actual test results.

17981 This work unit may be satisfied by provision of the test evidence from the evaluation of the
17982 dependent component if the base component was used to satisfy the requirements for IT in the
17983 operational environment of the dependent component.

17984 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

17985 a) that the test documentation consist of test plans expected test results and actual test
17986 results;

17987 b) that the test documentation contains the information necessary to ensure the tests are
17988 repeatable;

17989 c) the level of developer effort that was applied to testing of the base component.

17990 **17.6.1.3.2 Work unit ACO_CTT.1-2**

17991 The evaluator **shall examine** the base component interface test documentation to determine that it
17992 consists of test plans, expected test results and actual test results.

17993 This work unit may be satisfied by provision of the test evidence from the evaluation of the base
17994 component for those interfaces relied upon in the composed TOE by the dependent component are
17995 TSFIs of the successfully evaluated base component. The determination of whether the interfaces
17996 of the base component relied upon by the dependent component were in fact TSFIs of the
17997 evaluated base component is made during the ACO_COR activity.

17998 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

17999 a) that the test documentation consist of test plans expected test results and actual test
18000 results;

18001 b) that the test documentation contains the information necessary to ensure the tests are
18002 repeatable;

18003 c) the level of developer effort that was applied to testing of the base component.

18004 ISO/IEC 15408-3 ACO_CTT.1.2C: *The test documentation from the developer execution of the*
18005 *composed TOE tests shall demonstrate that the TSF behaves as specified.*

18006 **17.6.1.3.3 Work unit ACO_CTT.1-3**

18007 The evaluator **shall examine** the test documentation to determine that the developer execution of
18008 the composed TOE tests shall demonstrate that the TSF behaves as specified.

18009 The evaluator should construct a mapping between the tests described in the test plan and the
18010 SFRs specified for the composed TOE to identify which SFRs have been tested by the developer.

18011 Guidance on this work unit can be found in:

18012 a) Clause 15.2.2.

18013 b) Clause 15.2.3.

18014 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be
18015 compared with the mapping to determine that the SFRs of the composed TOE, as tested by the
18016 developer, behave as expected.

- 18017 ISO/IEC 15408-3 ACO_CTT.1.3C: *The test documentation from the developer execution of the base*
 18018 *component interface tests shall demonstrate that the base component interface relied upon by the*
 18019 *dependent component behaves as specified.*
- 18020 **17.6.1.3.4 Work unit ACO_CTT.1-4**
- 18021 The evaluator ***shall examine*** the test documentation to determine that the developer execution of
 18022 the base component interface tests shall demonstrate that the base component interfaces relied
 18023 upon by the dependent component behave as specified.
- 18024 The evaluator should construct a mapping between the tests described in the test plan and the
 18025 interfaces of the base component relied upon by the dependent component (as specified in the
 18026 reliance information, examined under ACO_REL) to identify which base component interfaces have
 18027 been tested by the developer.
- 18028 Guidance on this work unit can be found in:
- 18029 a) Clause 15.2.2.
- 18030 b) Clause 15.2.3.
- 18031 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be
 18032 compared with the mapping to determine that the interfaces of the base component, as tested by
 18033 the developer, behave as expected.
- 18034 ISO/IEC 15408-3 ACO_CTT.1.4C: *The base component shall be suitable for testing.*
- 18035 **17.6.1.3.5 Work unit ACO_CTT.1-5**
- 18036 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly
 18037 and is in a known state.
- 18038 To determine that the composed TOE has been installed properly and is in a known state the
 18039 ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the TOE provided by the developer for
 18040 testing.
- 18041 **17.6.1.3.6 Work unit ACO_CTT.1-6**
- 18042 The evaluator ***shall examine*** the set of resources provided by the developer to determine that they
 18043 are equivalent to the set of resources used by the base component developer to functionally test
 18044 the base component.
- 18045 To determine that the set of resources provided are equivalent to those used to functionally test
 18046 the base component as used in the composed TOE, the ATE_IND.2-3 work unit will be applied.
- 18047 **17.6.1.4 Action ACO_CTT.1.2E**
- 18048 **17.6.1.4.1 Work unit ACO_CTT.1-7**
- 18049 The evaluator ***shall perform*** testing in accordance with ATE_IND.2.2E, for a subset of the SFRs
 18050 specified in the composed security target, to verify the developer test results.
- 18051 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.2E, reporting in
 18052 the ETR for the composed TOE all analysis, results and verdicts as dictated by the associated work
 18053 units.

18054 **17.6.1.5 Action ACO_CTT.1.3E**

18055 **17.6.1.5.1 Work unit ACO_CTT.1-8**

18056 The evaluator **shall perform** testing in accordance with ATE_IND.2.3E, for a subset of the SFRs
18057 specified in the composed security target, to confirm that the TSF operates as specified.

18058 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in
18059 the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

18060 When selecting interfaces of the TSF of the composed TOE to test, the evaluator should take into
18061 account any modifications to the components from the evaluated version or configuration.
18062 Modifications to the component from that evaluated may include patches introduced, a different
18063 configuration as a result of modified guidance documentation, reliance an additional portion of the
18064 component that was not within the TSF of the component. These modifications will have been
18065 identified during the Composition rationale (ACO_COR) activity.

18066 **17.6.2 Evaluation of sub-activity (ACO_CTT.2)**

18067 **17.6.2.1 Objectives**

18068 The objective of this sub-activity is to determine whether the developer correctly performed and
18069 documented tests for each of the base component interfaces on which the dependent component
18070 relies. As part of this determination the evaluator repeats a sample of the tests performed by the
18071 developer and performs any additional tests required to fully demonstrate the expected behaviour
18072 of the composed TOE and the interfaces of the base component relied upon by the dependent
18073 component.

18074 **17.6.2.2 Input**

18075 The evaluation evidence for this sub-activity is:

- 18076 a) the composed TOE suitable for testing;
- 18077 b) the composed TOE testing evidence;
- 18078 c) the reliance information;
- 18079 d) the development information.

18080 **17.6.2.3 Action ACO_CTT.2.1E**

18081 ISO/IEC 15408-3 ACO_CTT.2.1C: *The composed TOE and base component interface test*
18082 *documentation shall consist of test plans, expected test results and actual test results.*

18083 **17.6.2.3.1 Work unit ACO_CTT.2-1**

18084 The evaluator **shall examine** the composed TOE test documentation to determine that it consists
18085 of test plans, expected test results and actual test results.

18086 This work unit may be satisfied by provision of the test evidence from the evaluation of the
18087 dependent component if the base component was used to satisfy the requirements for IT in the
18088 operational environment of the dependent component.

18089 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

- 18090 a) that the test documentation consist of test plans expected test results and actual test
18091 results;

18092 b) that the test documentation contains the information necessary to ensure the tests are
18093 repeatable;

18094 c) the level of developer effort that was applied to testing of the base component.

18095 **17.6.2.3.2 Work unit ACO_CTT.2-2**

18096 The evaluator ***shall examine*** the base component interface test documentation to determine that it
18097 consists of test plans, expected test results and actual test results.

18098 This work unit may be satisfied by provision of the test evidence from the evaluation of the base
18099 component for those interfaces relied upon in the composed TOE by the dependent component are
18100 TSFIs of the successfully evaluated base component. The determination of whether the interfaces
18101 of the base component relied upon by the dependent component were in fact TSFIs of the
18102 evaluated base component is made during the ACO_COR activity.

18103 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

18104 a) that the test documentation consist of test plans expected test results and actual test
18105 results;

18106 b) that the test documentation contains the information necessary to ensure the tests are
18107 repeatable;

18108 c) the level of developer effort that was applied to testing of the base component.

18109 ISO/IEC 15408-3 ACO_CTT.2.2C: *The test documentation from the developer execution of the*
18110 *composed TOE tests shall demonstrate that the TSF behaves as specified and is complete.*

18111 **17.6.2.3.3 Work unit ACO_CTT.2-3**

18112 The evaluator ***shall examine*** the test documentation to determine that it provides accurate
18113 correspondence between the tests in the test documentation relating to the testing of the
18114 composed TOE and the composed TOE SFRs in the composed TOE security target.

18115 A simple cross-table may be sufficient to show test correspondence. The identification of
18116 correspondence between the tests and SFRs presented in the test documentation has to be
18117 unambiguous.

18118 **17.6.2.3.4 Work unit ACO_CTT.2-4**

18119 The evaluator ***shall examine*** the test documentation to determine that the developer execution of
18120 the composed TOE tests shall demonstrate that the TSF behaves as specified.

18121 Guidance on this work unit can be found in:

18122 a) Clause 15.2.2.

18123 b) Clause 15.2.3.

18124 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be
18125 compared with the mapping to determine that the SFRs of the composed TOE, as tested by the
18126 developer, behave as expected.

18127 ISO/IEC 15408-3 ACO_CTT.2.3C: *The test documentation from the developer execution of the base*
18128 *component interface tests shall demonstrate that the base component interface relied upon by the*
18129 *dependent component behaves as specified and is complete.*

18130 **17.6.2.3.5 Work unit ACO_CTT.2-5**

18131 The evaluator *shall examine* the test documentation to determine that it provides accurate
18132 correspondence between the tests in the test documentation relating to the testing of the base
18133 component interfaces relied upon by the dependent component and the interfaces specified in the
18134 reliance information.

18135 A simple cross-table may be sufficient to show test correspondence. The identification of
18136 correspondence between the tests and interfaces presented in the test documentation has to be
18137 unambiguous.

18138 **17.6.2.3.6 Work unit ACO_CTT.2-6**

18139 The evaluator *shall examine* the test documentation to determine that the developer execution of
18140 the base component interface tests shall demonstrate that the base component interfaces relied
18141 upon by the dependent component behave as specified.

18142 Guidance on this work unit can be found in:

18143 a) Clause 15.2.2.

18144 b) Clause 15.2.3.

18145 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be
18146 compared with the mapping to determine that the interfaces of the base component, as tested by
18147 the developer, behave as expected.

18148 ISO/IEC 15408-3 ACO_CTT.2.4C: *The base component shall be suitable for testing.*

18149 **17.6.2.3.7 Work unit ACO_CTT.2-7**

18150 The evaluator *shall examine* the composed TOE to determine that it has been installed properly
18151 and is in a known state.

18152 To determine that the composed TOE has been installed properly and is in a known state the
18153 ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the TOE provided by the developer for
18154 testing.

18155 **17.6.2.3.8 Work unit ACO_CTT.2-8**

18156 The evaluator *shall examine* the set of resources provided by the developer to determine that they
18157 are equivalent to the set of resources used by the base component developer to functionally test
18158 the base component.

18159 To determine that the set of resources provided are equivalent to those used to functionally test
18160 the base component as used in the composed TOE, the ATE_IND.2-3 work unit will be applied.

18161 **17.6.2.4 Action ACO_CTT.2.2E**

18162 **17.6.2.4.1 Work unit ACO_CTT.2-9**

18163 The tests are to be selected and executed in accordance with ATE_IND.2.2E, to demonstrate the
18164 correct behaviour of the SFRs specified in the composed TOE security target.

18165 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.2E, reporting in
18166 the ETR for the composed TOE all analysis, results and verdicts as dictated by the associated work
18167 units.

18168 **17.6.2.5 Action ACO_CTT.2.3E**18169 **17.6.2.5.1 Work unit ACO_CTT.2-10**

18170 The evaluator ***shall perform*** testing in accordance with ATE_IND.2.3E, for a subset of the SFRs
 18171 specified in the composed security target, to confirm that the TSF operates as specified.

18172 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in
 18173 the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

18174 When selecting interfaces of the TSF of the composed TOE to test, the evaluator should take into
 18175 account any modifications to the components from the evaluated version or configuration.
 18176 Modifications to the component from that evaluated may include patches introduced, a different
 18177 configuration as a result of modified guidance documentation, reliance an additional portion of the
 18178 component that was not within the TSF of the component. These modifications will have been
 18179 identified during the Composition rationale (ACO_COR) activity.

18180 **17.6.2.5.2 Work unit ACO_CTT.2-11**

18181 The evaluator ***shall perform*** testing, in accordance with Evaluation of sub-activity (ATE_IND.2), for
 18182 a subset of the interfaces to the base component to confirm they operate as specified.

18183 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in
 18184 the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

18185 When selecting interfaces of the base component to test, the evaluator should take into account any
 18186 modifications to the base component from the evaluated version or configuration. In particular, the
 18187 evaluator should consider the development of tests to demonstrate the correct behaviour of
 18188 interfaces of the base component that were not considered during the evaluation of the base
 18189 component. These additional interfaces and other modifications to the base component will have
 18190 been identified during the Composition rationale (ACO_COR) activity.

18191 **17.7 Composition vulnerability analysis (ACO_VUL)**18192 **17.7.1 Evaluation of sub-activity (ACO_VUL.1)**18193 **17.7.1.1 Objectives**

18194 The objective of this sub-activity is to determine whether the composed TOE, in its operational
 18195 environment, has easily exploitable vulnerabilities.

18196 The developer provides details of any residual vulnerabilities reported from evaluation of the
 18197 components. The evaluator performs an analysis of the disposition the residual vulnerabilities
 18198 reported and also performs a search of the public domain, to identify any new potential
 18199 vulnerabilities in the components (i.e. those issues that have been reported in the public domain
 18200 since evaluation of the base component). The evaluator then performs penetration testing to
 18201 demonstrate that the potential vulnerabilities cannot be exploited in the TOE, in its operational
 18202 environment, by an attacker with basic attack potential.

18203 **17.7.1.2 Input**

18204 The evaluation evidence for this sub-activity is:

18205 a) the composed TOE suitable for testing;

18206 b) the composed ST;

18207 c) the composition rationale;

- 18208 d) the guidance documentation;
- 18209 e) information publicly available to support the identification of possible security
18210 vulnerabilities;
- 18211 f) residual vulnerabilities reported during evaluation of each component.
- 18212 **17.7.1.3 Application notes**
- 18213 See the application notes for Evaluation of sub-activity (AVA_VAN.1).
- 18214 **17.7.1.4 Action ACO_VUL.1.1E**
- 18215 ISO/IEC 15408-3 ACO_VUL.1.1C: *The composed TOE shall be suitable for testing.*
- 18216 **17.7.1.4.1 Work unit ACO_VUL.1-1**
- 18217 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly
18218 and is in a known state.
- 18219 To determine that the composed TOE has been installed properly and is in a known state the
18220 ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.
- 18221 If the assurance package includes a component from the ACO_CTT family, then the evaluator may
18222 refer to the result of the work unit ACO_CTT*-1 to demonstrate this has been satisfied.
- 18223 **17.7.1.4.2 Work unit ACO_VUL.1-2**
- 18224 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions
18225 and objectives in the STs the components relating to IT entities for are fulfilled by the other
18226 components.
- 18227 The STs for the component may include assumptions about other components that may use the
18228 component to which the ST relates, e.g. the ST for an operating system used as a base component
18229 may include an assumption that any applications loaded on the operating system do not run in
18230 privileged mode. These assumptions and objectives are to be fulfilled by other components in the
18231 composed TOE.
- 18232 **17.7.1.5 Action ACO_VUL.1.2E**
- 18233 **17.7.1.5.1 Work unit ACO_VUL.1-3**
- 18234 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to
18235 determine that they are not exploitable in the composed TOE in its operational environment.
- 18236 The list of vulnerabilities identified in the product during the evaluation of the base component,
18237 which were demonstrated to be non-exploitable in the base component, is to be used as an input
18238 into this activity. The evaluator will determine that the premise(s) on which a vulnerability was
18239 deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-
18240 introduced the potential vulnerability. For example, if during evaluation of the base component it
18241 was assumed that a particular operating system service was disabled, which is enabled in the
18242 composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped
18243 out should now be considered.
- 18244 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base
18245 component should be considered in the light of any known, non-exploitable vulnerabilities for the
18246 other components (e.g. dependent component) within the composed TOE. This is to consider the

18247 case where a potential vulnerability that is non-exploitable in isolation is exploitable when
18248 integrated with an IT entity containing another potential vulnerability.

18249 **17.7.1.5.2 Work unit ACO_VUL.1-4**

18250 The evaluator *shall examine* the residual vulnerabilities from the dependent component
18251 evaluation to determine that they are not exploitable in the composed TOE in its operational
18252 environment.

18253 The list of vulnerabilities identified in the product during the evaluation of the dependent
18254 component, which were demonstrated to be non-exploitable in the dependent component, is to be
18255 used as an input into this activity. The evaluator will determine that the premise(s) on which a
18256 vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the
18257 combination has re-introduced the potential vulnerability. For example, if during evaluation of the
18258 dependent component it was assumed that IT meeting the operational environment requirements
18259 would not return a certain value in response to a service request, which is provided by the base
18260 component in the composed TOE evaluation, any potential vulnerabilities relating to that return
18261 value previously scoped out should now be considered.

18262 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the
18263 dependent component should be considered in the light of any known, non-exploitable
18264 vulnerabilities for the other components (e.g. base component) within the composed TOE. This is
18265 to consider the case where a potential vulnerability that is non-exploitable in isolation is
18266 exploitable when integrated with an IT entity containing another potential vulnerability.

18267 **17.7.1.6 Action ACO_VUL.1.3E**

18268 **17.7.1.6.1 Work unit ACO_VUL.1-5**

18269 The evaluator *shall examine* the sources of information publicly available to support the
18270 identification of possible security vulnerabilities in the base component that have become known
18271 since the completion of evaluation of the base component.

18272 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search
18273 for vulnerabilities in the base component.

18274 Those potential vulnerabilities that were publicly available prior to the evaluation of the base
18275 component do not have to be further investigated unless it is apparent to the evaluator that the
18276 attack potential required by an attacker to exploit the potential vulnerability has been significantly
18277 reduced. This may be through the introduction of some new technology since the base component
18278 evaluation that means the exploitation of the potential vulnerability has been simplified.

18279 **17.7.1.6.2 Work unit ACO_VUL.1-6**

18280 The evaluator *shall examine* the sources of information publicly available to support the
18281 identification of possible security vulnerabilities in the dependent component that have become
18282 known since the completion of the dependent component evaluation.

18283 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search
18284 for vulnerabilities in the dependent component.

18285 Those potential vulnerabilities that were publicly available prior to the evaluation of the
18286 dependent component do not have to be further investigated unless it is apparent to the evaluator
18287 that the attack potential required by an attacker to exploit the potential vulnerability has been
18288 significantly reduced. This may be through the introduction of some new technology since
18289 evaluation of the dependent component that means the exploitation of the potential vulnerability
18290 has been simplified.

18291 **17.7.1.6.3 Work unit ACO_VUL.1-7**

18292 The evaluator **shall record** in the ETR the identified potential security vulnerabilities that are
18293 candidates for testing and applicable to the composed TOE in its operational environment.

18294 The ST, guidance documentation and functional specification are used to determine whether the
18295 vulnerabilities are relevant to the composed TOE in its operational environment.

18296 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the
18297 evaluator determines that the vulnerability is not applicable in the operational environment.
18298 Otherwise the evaluator records the potential vulnerability for further consideration.

18299 A list of potential vulnerabilities applicable to the composed TOE in its operational environment,
18300 which can be used as an input into penetration testing activities (i.e. ACO_VUL.1.4E), shall be
18301 reported in the ETR by the evaluators.

18302 **17.7.1.7 Action ACO_VUL.1.4E**

18303 **17.7.1.7.1 Work unit ACO_VUL.1-8**

18304 The evaluator **shall conduct** penetration testing as detailed for AVA_VAN.1.3E.

18305 The evaluator will apply all work units necessary for the satisfaction of evaluator action
18306 AVA_VAN.1.3E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by
18307 the work units.

18308 The evaluator will also apply the work units for the evaluator action AVA_VAN.1.1E to determine
18309 that the composed TOE provided by the developer is suitable for testing.

18310 **17.7.2 Evaluation of sub-activity (ACO_VUL.2)**

18311 **17.7.2.1 Objectives**

18312 The objective of this sub-activity is to determine whether the composed TOE, in its operational
18313 environment, has vulnerabilities exploitable by attackers possessing basic attack potential.

18314 The developer provides an analysis of the disposition of any residual vulnerabilities reported for
18315 the components and of any vulnerabilities introduced through the combination of the base and
18316 dependent components. The evaluator performs a search of the public domain to identify any new
18317 potential vulnerabilities in the components (i.e. those issues that have been reported in the public
18318 domain since the completion of the evaluation of the components). The evaluator will also perform
18319 an independent vulnerability analysis of the composed TOE and penetration testing.

18320 **17.7.2.2 Input**

18321 The evaluation evidence for this sub-activity is:

18322 a) the composed TOE suitable for testing;

18323 b) the composed ST;

18324 c) the composition rationale;

18325 d) the reliance information;

18326 e) the guidance documentation;

- 18327 f) information publicly available to support the identification of possible security
18328 vulnerabilities.
- 18329 g) residual vulnerabilities reported during evaluation of each component.
- 18330 **17.7.2.3 Application notes**
- 18331 See the application notes for Evaluation of sub-activity (AVA_VAN.2).
- 18332 **17.7.2.4 Action ACO_VUL.2.1E**
- 18333 ISO/IEC 15408-3 ACO_VUL.2.1C: *The composed TOE shall be suitable for testing.*
- 18334 **17.7.2.4.1 Work unit ACO_VUL.2-1**
- 18335 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly
18336 and is in a known state.
- 18337 To determine that the composed TOE has been installed properly and is in a known state the
18338 ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.
- 18339 If the assurance package includes ACO_CTT family, then the evaluator may refer to the result of the
18340 work unit Composed TOE testing (ACO_CTT)*-1 to demonstrate this has been satisfied.
- 18341 **17.7.2.4.2 Work unit ACO_VUL.2-2**
- 18342 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions
18343 and objectives in the STs the components relating to IT entities for are fulfilled by the other
18344 components.
- 18345 The STs for the component may include assumptions about other components that may use the
18346 component to which the ST relates, e.g. the ST for an operating system used as a base component
18347 may include an assumption that any applications loaded on the operating system do not run in
18348 privileged mode. These assumptions and objectives are to be fulfilled by other components in the
18349 composed TOE.
- 18350 **17.7.2.5 Action ACO_VUL.2.2E**
- 18351 **17.7.2.5.1 Work unit ACO_VUL.2-3**
- 18352 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to
18353 determine that they are not exploitable in the composed TOE in its operational environment.
- 18354 The list of vulnerabilities identified in the product during the evaluation of the base component,
18355 which were demonstrated to be non-exploitable in the base component, is to be used as an input
18356 into this activity. The evaluator will determine that the premise(s) on which a vulnerability was
18357 deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-
18358 introduced the potential vulnerability. For example, if during evaluation of the base component it
18359 was assumed that a particular operating system service was disabled, which is enabled in the
18360 composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped
18361 out should now be considered.
- 18362 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base
18363 component should be considered in the light of any known, non-exploitable vulnerabilities for the
18364 other components (e.g. dependent component) within the composed TOE. This is to consider the
18365 case where a potential vulnerability that is non-exploitable in isolation is exploitable when
18366 integrated with an IT entity containing another potential vulnerability.

18367 **17.7.2.5.2 Work unit ACO_VUL.2-4**

18368 The evaluator *shall examine* the residual vulnerabilities from the dependent component
18369 evaluation to determine that they are not exploitable in the composed TOE in its operational
18370 environment.

18371 The list of vulnerabilities identified in the product during the evaluation of the dependent
18372 component, which were demonstrated to be non-exploitable in the dependent component, is to be
18373 used as an input into this activity. The evaluator will determine that the premise(s) on which a
18374 vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the
18375 combination has re-introduced the potential vulnerability. For example, if during evaluation of the
18376 dependent component it was assumed that IT meeting the operational environment requirements
18377 would not return a certain value in response to a service request, which is provided by the base
18378 component in the composed TOE evaluation, any potential vulnerabilities relating to that return
18379 value previously scoped out should now be considered.

18380 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the
18381 dependent component should be considered in the light of any known, non-exploitable
18382 vulnerabilities for the other components (e.g. base component) within the composed TOE. This is
18383 to consider the case where a potential vulnerability that is non-exploitable in isolation is
18384 exploitable when integrated with an IT entity containing another potential vulnerability.

18385 **17.7.2.6 Action ACO_VUL.2.3E**

18386 **17.7.2.6.1 Work unit ACO_VUL.2-5**

18387 The evaluator *shall examine* the sources of information publicly available to support the
18388 identification of possible security vulnerabilities in the base component that have become known
18389 since the completion of the base component evaluation.

18390 The evaluator will use the information in the public domain as described in AVA_VAN.2-2 to search
18391 for vulnerabilities in the base component.

18392 Those potential vulnerabilities that were publicly available prior to the evaluation of the base
18393 component do not have to be further investigated unless it is apparent to the evaluator that the
18394 attack potential required by an attacker to exploit the potential vulnerability has been significantly
18395 reduced. This may be through the introduction of some new technology since the base component
18396 evaluation that means the exploitation of the potential vulnerability has been simplified.

18397 **17.7.2.6.2 Work unit ACO_VUL.2-6**

18398 The evaluator *shall examine* the sources of information publicly available to support the
18399 identification of possible security vulnerabilities in the dependent component that have become
18400 known since the completion of the dependent component evaluation.

18401 The evaluator will use the information in the public domain as described in AVA_VAN.2-2 to search
18402 for vulnerabilities in the dependent component.

18403 Those potential vulnerabilities that were publicly available prior to the evaluation of the
18404 dependent component do not have to be further investigated unless it is apparent to the evaluator
18405 that the attack potential required by an attacker to exploit the potential vulnerability has been
18406 significantly reduced. This may be through the introduction of some new technology since
18407 evaluation of the dependent component that means the exploitation of the potential vulnerability
18408 has been simplified.

18409 **17.7.2.6.3 Work unit ACO_VUL.2-7**

18410 The evaluator **shall record** in the ETR the identified potential security vulnerabilities that are
18411 candidates for testing and applicable to the composed TOE in its operational environment.

18412 The ST, guidance documentation and functional specification are used to determine whether the
18413 vulnerabilities are relevant to the composed TOE in its operational environment.

18414 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the
18415 evaluator determines that the vulnerability is not applicable in the operational environment.
18416 Otherwise the evaluator records the potential vulnerability for further consideration.

18417 A list of potential vulnerabilities applicable to the composed TOE in its operational environment,
18418 which can be used as an input into penetration testing activities (ACO_VUL.2.5E), shall be reported
18419 in the ETR by the evaluators.

18420 **17.7.2.7 Action ACO_VUL.2.4E**18421 **17.7.2.7.1 Work unit ACO_VUL.2-8**

18422 The evaluator **shall conduct** a search of the composed TOE ST, guidance documentation, reliance
18423 information and composition rationale to identify possible security vulnerabilities in the composed
18424 TOE.

18425 The consideration of the components of the composed TOE in the independent evaluator
18426 vulnerability analysis will take a slightly different form to that documented in AVA_VAN.2.3E for a
18427 component evaluation, as it will not necessarily consider all layers of design abstraction relevant to
18428 the assurance package. These will have already been considered during the evaluation of the
18429 components, but the evidence may not be available for the composed TOE evaluation. However, the
18430 general approach described in the work units associated with AVA_VAN.2.3E is applicable and
18431 should form the basis of the evaluator's search for potential vulnerabilities in the composed TOE.

18432 A vulnerability analysis of the individual components used in the composed TOE will have already
18433 been performed during evaluation of the individual components. The focus of the vulnerability
18434 analysis during the composed TOE evaluation is to identify any vulnerabilities introduced as a
18435 result of the integration of the components or due to any changes in the use of the components
18436 between the evaluated component configuration to the composed TOE configuration.

18437 The evaluator will use the understanding of the component's construction as detailed in the
18438 reliance information for the dependent component, and the development information and
18439 composition rationale for the base component, together with the dependent component design
18440 information. This information will allow the evaluator to gain an understanding of how the base
18441 component and dependent component interact and identify potential vulnerabilities that may be
18442 introduced as a result of this interaction.

18443 The evaluator will consider any new guidance provided for the installation, start-up and operation
18444 of the composed TOE to identify any potential vulnerabilities introduced through this revised
18445 guidance.

18446 If any of the individual components have been through assurance continuity activities since the
18447 completion of the component evaluation, the evaluator will consider the patch(es) in the
18448 independent vulnerability analysis. Information related to the change provided in a public report of
18449 the assurance continuity activities (e.g. Maintenance Report) will be the main source of input
18450 material of the change. This will be supplemented by any updates to the guidance documentation
18451 resulting from the change and any information regarding the change available in the public domain,
18452 e.g. vendor website.

18453 Any risks identified due to the lack of evidence to establish the full impact of any patches or
18454 deviations in the configuration of a component from the evaluated configuration are to be
18455 documented in the evaluator's vulnerability analysis.

18456 **17.7.2.8 Action ACO_VUL.2.5E**

18457 **17.7.2.8.1 Work unit ACO_VUL.2-9**

18458 The evaluator **shall conduct** penetration testing as detailed for AVA_VAN.2.4E.

18459 The evaluator will apply all work units necessary for the satisfaction of evaluator action
18460 AVA_VAN.2.4E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by
18461 the work units.

18462 The evaluator will also apply the work units for the evaluator action AVA_VAN.2.1E to determine
18463 that the composed TOE provided by the developer is suitable for testing.

18464 **17.7.3 Evaluation of sub-activity (ACO_VUL.3)**

18465 **17.7.3.1 Objectives**

18466 The objective of this sub-activity is to determine whether the composed TOE, in its operational
18467 environment, has vulnerabilities exploitable by attackers possessing Enhanced-Basic attack
18468 potential.

18469 The developer provides an analysis of the disposition of any residual vulnerabilities reported for
18470 the components and of any vulnerabilities introduced through the combination of the base and
18471 dependent components. The evaluator performs a search of the public domain to identify any new
18472 potential vulnerabilities in the components (i.e. those issues that have been reported in the public
18473 domain since the completion of the component evaluations). The evaluator will also perform an
18474 independent vulnerability analysis of the composed TOE and penetration testing.

18475 **17.7.3.2 Input**

18476 The evaluation evidence for this sub-activity is:

18477 a) the composed TOE suitable for testing;

18478 b) the composed ST;

18479 c) the composition rationale;

18480 d) the reliance information;

18481 e) the guidance documentation;

18482 f) information publicly available to support the identification of possible security
18483 vulnerabilities.

18484 g) residual vulnerabilities reported during evaluation of each component.

18485 **17.7.3.3 Application notes**

18486 See the application notes for Evaluation of sub-activity (AVA_VAN.3).

18487 **17.7.3.4 Action ACO_VUL.3.1E**

18488 ISO/IEC 15408-3 ACO_VUL.3.1C: *The composed TOE shall be suitable for testing.*

18489 **17.7.3.4.1 Work unit ACO_VUL.3-1**

18490 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly
18491 and is in a known state.

18492 To determine that the composed TOE has been installed properly and is in a known state the
18493 ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.

18494 If the assurance package includes ACO_CTT family, then the evaluator may refer to the result of the
18495 work unit Composed TOE testing (ACO_CTT)*-1 to demonstrate this has been satisfied.

18496 **17.7.3.4.2 Work unit ACO_VUL.3-2**

18497 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions
18498 and objectives in the STs the components relating to IT entities for are fulfilled by the other
18499 components.

18500 The STs for the component may include assumptions about other components that may use the
18501 component to which the ST relates, e.g. the ST for an operating system used as a base component
18502 may include an assumption that any applications loaded on the operating system do not run in
18503 privileged mode. These assumptions and objectives are to be fulfilled by other components in the
18504 composed TOE.

18505 **17.7.3.5 Action ACO_VUL.3.2E**18506 **17.7.3.5.1 Work unit ACO_VUL.3-3**

18507 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to
18508 determine that they are not exploitable in the composed TOE in its operational environment.

18509 The list of vulnerabilities identified in the product during the evaluation of the base component,
18510 which were demonstrated to be non-exploitable in the base component, is to be used as an input
18511 into this activity. The evaluator will determine that the premise(s) on which a vulnerability was
18512 deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-
18513 introduced the potential vulnerability. For example, if during evaluation of the base component it
18514 was assumed that a particular operating system service was disabled, which is enabled in the
18515 composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped
18516 out should now be considered.

18517 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base
18518 component should be considered in the light of any known, non-exploitable vulnerabilities for the
18519 other components (e.g. dependent component) within the composed TOE. This is to consider the
18520 case where a potential vulnerability that is non-exploitable in isolation is exploitable when
18521 integrated with an IT entity containing another potential vulnerability.

18522 **17.7.3.5.2 Work unit ACO_VUL.3-4**

18523 The evaluator ***shall examine*** the residual vulnerabilities from the dependent component
18524 evaluation to determine that they are not exploitable in the composed TOE in its operational
18525 environment.

18526 The list of vulnerabilities identified in the product during the evaluation of the dependent
18527 component, which were demonstrated to be non-exploitable in the dependent component, is to be
18528 used as an input into this activity. The evaluator will determine that the premise(s) on which a
18529 vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the
18530 combination has re-introduced the potential vulnerability. For example, if during evaluation of the
18531 dependent component it was assumed that IT meeting the operational environment requirements
18532 would not return a certain value in response to a service request, which is provided by the base

18533 component in the composed TOE evaluation, any potential vulnerabilities relating to that return
18534 value previously scoped out should now be considered.

18535 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the
18536 dependent component should be considered in the light of any known, non-exploitable
18537 vulnerabilities for the other components (e.g. base component) within the composed TOE. This is
18538 to consider the case where a potential vulnerability that is non-exploitable in isolation is
18539 exploitable when integrated with an IT entity containing another potential vulnerability.

18540 **17.7.3.6 Action ACO_VUL.3.3E**

18541 **17.7.3.6.1 Work unit ACO_VUL.3-5**

18542 The evaluator ***shall examine*** the sources of information publicly available to support the
18543 identification of possible security vulnerabilities in the base component that have become known
18544 since the completion of the base component evaluation.

18545 The evaluator will use the information in the public domain as described in AVA_VAN.3-2 to search
18546 for vulnerabilities in the base component.

18547 Those potential vulnerabilities that were publicly available prior to the evaluation of the base
18548 component do not have to be further investigated unless it is apparent to the evaluator that the
18549 attack potential required by an attacker to exploit the potential vulnerability has been significantly
18550 reduced. This may be through the introduction of some new technology since the base component
18551 evaluation that means the exploitation of the potential vulnerability has been simplified.

18552 **17.7.3.6.2 Work unit ACO_VUL.3-6**

18553 The evaluator ***shall examine*** the sources of information publicly available to support the
18554 identification of possible security vulnerabilities in the dependent component that have become
18555 known since completion of the dependent component evaluation.

18556 The evaluator will use the information in the public domain as described in AVA_VAN.3-2 to search
18557 for vulnerabilities in the dependent component.

18558 Those potential vulnerabilities that were publicly available prior to the evaluation of the
18559 dependent component do not have to be further investigated unless it is apparent to the evaluator
18560 that the attack potential required by an attacker to exploit the potential vulnerability has been
18561 significantly reduced. This may be through the introduction of some new technology since
18562 evaluation of the dependent component that means the exploitation of the potential vulnerability
18563 has been simplified.

18564 **17.7.3.6.3 Work unit ACO_VUL.3-7**

18565 The evaluator ***shall record*** in the ETR the identified potential security vulnerabilities that are
18566 candidates for testing and applicable to the composed TOE in its operational environment.

18567 The ST, guidance documentation and functional specification are used to determine whether the
18568 vulnerabilities are relevant to the composed TOE in its operational environment.

18569 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the
18570 evaluator determines that the vulnerability is not applicable in the operational environment.
18571 Otherwise the evaluator records the potential vulnerability for further consideration.

18572 A list of potential vulnerabilities applicable to the composed TOE in its operational environment,
18573 which can be used as an input into penetration testing activities (ACO_VUL.3.5E), shall be reported
18574 in the ETR by the evaluators.

18575 **17.7.3.7 Action ACO_VUL.3.4E**18576 **17.7.3.7.1 Work unit ACO_VUL.3-8**

18577 The evaluator ***shall conduct*** a search of the composed TOE ST, guidance documentation, reliance
 18578 information and composition rationale to identify possible security vulnerabilities in the composed
 18579 TOE.

18580 The consideration of the components in the independent evaluator vulnerability analysis will take
 18581 a slightly different form to that documented in AVA_VAN.3.3E for a component evaluation, as it will
 18582 not necessarily consider all layers of design abstraction relevant to the assurance package. These
 18583 will have already been considered during the evaluation of the base component, but the evidence
 18584 may not be available for the composed TOE evaluation. However, the general approach described
 18585 in the work units associated with AVA_VAN.3.3E is applicable and should form the basis of the
 18586 evaluator's search for potential vulnerabilities in the composed TOE.

18587 A vulnerability analysis of the individual components used in the composed TOE will have already
 18588 been performed during evaluation of the components. The focus of the vulnerability analysis
 18589 during the composed TOE evaluation is to identify any vulnerabilities introduced as a result of the
 18590 integration of the components or due to any changes in the use of the components between the
 18591 configuration of the component determined during the component evaluation and the composed
 18592 TOE configuration.

18593 The evaluator will use the understanding of the component's construction as detailed in the
 18594 reliance information for the dependent component, and the composition rationale and
 18595 development information for the base component, together with the dependent component design
 18596 information. This information will allow the evaluator to gain an understanding of how the base
 18597 component and dependent component interact.

18598 The evaluator will consider any new guidance provided for the installation, start-up and operation
 18599 of the composed TOE to identify any potential vulnerabilities introduced through this revised
 18600 guidance.

18601 If any of the individual components have been through assurance continuity activities since the
 18602 completion of the component evaluation, the evaluator will consider the patch in the independent
 18603 vulnerability analysis. Information related to the change provided in a public report of the
 18604 assurance continuity activities (e.g. Maintenance Report). This will be supplemented by any
 18605 updates to the guidance documentation resulting from the change and any information regarding
 18606 the change available in the public domain, e.g. vendor website.

18607 Any risks identified due to the lack of evidence to establish the full impact of any patches or
 18608 deviations in the configuration of a component from the evaluated configuration are to be
 18609 documented in the evaluator's vulnerability analysis.

18610 **17.7.3.8 Action ACO_VUL.3.5E**18611 **17.7.3.8.1 Work unit ACO_VUL.3-9**

18612 The evaluator ***shall conduct*** penetration testing as detailed for AVA_VAN.3.4E.

18613 The evaluator will apply all work units necessary for the satisfaction of evaluator action
 18614 AVA_VAN.3.4E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by
 18615 the work units.

18616 The evaluator will also apply the work units for the evaluator action AVA_VAN.3.1E to determine
 18617 that the composed TOE provided by the developer is suitable for testing.

Annex A (informative)

General evaluation guidance

A.1 Objectives

The objective of this clause is to cover general guidance used to provide technical evidence of evaluation results. The use of such general guidance helps in achieving objectivity, repeatability and reproducibility of the work performed by the evaluator.

A.2 Sampling

This Subclause provides general guidance on sampling. Specific and detailed information is given in those work units under the specific evaluator action elements where sampling has to be performed.

Sampling is a defined procedure of an evaluator whereby some subset of a required set of evaluation evidence is examined and assumed to be representative for the entire set. It allows the evaluator to gain enough confidence in the correctness of particular evaluation evidence without analysing the whole evidence. The reason for sampling is to conserve resources while maintaining an adequate level of assurance. Sampling of the evidence can provide two possible outcomes:

- a) The subset reveals no errors, allowing the evaluator to have some confidence that the entire set is correct.
- b) The subset reveals errors and therefore the validity of the entire set is called into question. Even the resolution of all errors that were found may be insufficient to provide the evaluator the necessary confidence and as a result the evaluator may have to increase the size of the subset, or stop using sampling for this particular evidence.

Sampling is a technique which can be used to reach a reliable conclusion if a set of evidence is relatively homogeneous in nature, e.g. if the evidence has been produced during a well defined process.

Sampling in the cases identified in ISO/IEC 15408, and in cases specifically covered in evaluation methodology work items, is recognised as a cost-effective approach to performing evaluator actions. Sampling in other areas is permitted only in exceptional cases, where performance of a particular activity in its entirety would require effort disproportionate to the other evaluation activities, and where this would not add correspondingly to assurance. In such cases a rationale for the use of sampling in that area will need to be made. Neither the fact that the TOE is large and complex, nor that it has many security functional requirements, is sufficient justification, since evaluations of large, complex TOEs can be expected to require more effort. Rather it is intended that this exception be limited to cases such as that where the TOE development approach yields large quantities of material for a particular ISO/IEC 15408 requirement that would normally all need to be checked or examined, and where such an action would not be expected to raise assurance correspondingly.

Sampling needs to be justified taking into account the possible impact on the security objectives and threats of the TOE. The impact depends on what might be missed as a result of sampling. Consideration also needs to be given to the nature of the evidence to be sampled, and the requirement not to diminish or ignore any security functions.

It should be recognised that sampling of evidence directly related to the implementation of the TOE (e.g. developer test results) requires a different approach to sampling, then sampling related to the

determination of whether a process is being followed. In many cases the evaluator is required to determine that a process is being followed, and a sampling strategy is recommended. The approach for sampling a developer's test results will differ. This is because the former case is concerned with ensuring that a process is in place, and the latter deals with determining correct implementation of the TOE. Typically, larger sample sizes should be analysed in cases related to the correct implementation of the TOE than would be necessary to ensure that a process is in place.

17.7.3.9 In certain cases it may be appropriate for the evaluator to give greater emphasis to the repetition of developer testing. For example if the independent tests left for the evaluator to perform would be only superficially different from those included in an extensive developer test set (possibly because the developer has performed more testing than necessary to satisfy the Coverage (ATE_COV) and Objectives

17.7.3.9 The objective of this sub-activity is to determine whether the developer has tested all of the TSFIs exhaustively, and that the developer's test coverage evidence shows correspondence between the tests identified in the test documentation and the TSFIs described in the functional specification.

17.7.3.9 A particular objective of this component is to confirm that all parameters of all of the TSFIs have been tested.

17.7.3.9 Input

17.7.3.9 The evaluation evidence for this sub-activity is:

17.7.3.9 the ST;

17.7.3.9 the functional specification;

17.7.3.9 the test documentation;

17.7.3.9 the test coverage analysis.

17.7.3.9 Action ATE_COV.3.1E

17.7.3.9 ISO/IEC 15408-3 ATE_COV.3.1C: *The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.*

17.7.3.9 Work unit ATE_COV.3-1

17.7.3.9 The evaluator *shall examine* the test coverage analysis to determine that the correspondence between the tests in the test documentation and the interfaces in the functional specification is accurate.

17.7.3.9 A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the interfaces presented in the test coverage analysis has to be unambiguous.

17.7.3.9 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the functional specification.

17.7.3.9 Work unit ATE_COV.3-2

17.7.3.9 The evaluator *shall examine* the test plan to determine that the testing approach for each interface demonstrates the expected behaviour of that interface.

17.7.3.9 Guidance on this work unit can be found in:

17.7.3.9 15.2.1 Understanding the expected behaviour of the TOE

- 18700 **17.7.3.9 15.2.2 [Testing vs. alternate approaches to verify the expected behaviour of**
18701 **functionality**
- 18702 **17.7.3.9 Work unit ATE_COV.3-3**
- 18703 **17.7.3.9** The evaluator *shall examine the test procedures to determine that the test*
18704 *prerequisites, test steps and expected result(s) adequately test each interface.*
- 18705 **17.7.3.9** Guidance on this work units, as it pertains to the functional specification, can be found
18706 *in:*
- 18707 **17.7.3.9 15.2.3 Verifying the adequacy of tests**
- 18708 **17.7.3.9** ISO/IEC 15408-3 ATE_COV.3.2C *The analysis of the test coverage shall demonstrate*
18709 *that all TSFIs in the functional specification have been completely tested.*
- 18710 **17.7.3.9 Work unit ATE_COV.3-4**
- 18711 **17.7.3.9** The evaluator *shall examine the test coverage analysis to determine that the*
18712 *correspondence between the interfaces in the functional specification and the tests in the test*
18713 *documentation is complete.*
- 18714 **17.7.3.9** All TSFIs that are described in the functional specification have to be present in the test
18715 *coverage analysis and mapped to tests in order for completeness to be claimed. Exhaustive*
18716 *specification testing of interfaces is required for this mapping. Incomplete coverage would be*
18717 *evident if an interface was identified in the functional specification and no test was mapped to it.*
- 18718 **17.7.3.9** The evaluator is reminded that this does not imply that all tests in the test
18719 *documentation must map to interfaces in the functional specification.*
- 18720 **17.7.3.9 Work unit ATE_COV.3-5**
- 18721 **17.7.3.9** The evaluator *shall examine the test coverage analysis to determine that the*
18722 *correspondence between the interfaces in the functional specification and the tests in the test*
18723 *documentation shows that all TSFIs were tested completely.*
- 18724 **17.7.3.9** This means that the evaluator examines whether all aspects of purpose, method of use,
18725 *parameters, parameter descriptions, actions and error messages for all TSFIs present in the*
18726 *functional specification are covered by the tests. Note that the level of detail present in the*
18727 *functional specification depends on the component of ADV_FSP chosen in the ST of the TOE.*
- 18728 **17.7.3.9** The evaluator may conclude that the higher level descriptions in the functional
18729 *specification, like purpose or method of use, are implicitly covered, if coverage of lower level*
18730 *descriptions like parameters, parameter descriptions, actions and error messages are covered.*
18731 *Therefore in general it will only be necessary to confirm coverage on these lower levels.*
- 18732 **17.7.3.9** The evaluator is reminded that (for example) coverage of all parameters does not
18733 *necessarily mean coverage of every possible value a parameter may allow. However every value*
18734 *for which a distinct qualitative behaviour of the TOE is expected, needs to be covered.*
- 18735 **17.7.3.9** As an example: If one of the parameters of a function call is a two byte value, which
18736 *specifies the length of further parameters, only some typical values need to be tested. However*
18737 *the evaluator will make sure that some specific cases (like the value zero or the maximal value)*
18738 *will be covered.*
- 18739 **17.7.3.9** If the evaluator sees that a potential attacker might be able to invoke a TSFI with
18740 *inconsistent parameter values (e. g. if one parameter specifies the length of a second parameter*
18741 *and it is possible to make the second parameter actually longer than the chosen value for the first*

- 18742 parameter suggests) and this case is not covered by the developer's testing, the evaluator may
 18743 decide either to test this during their **activities in AVA_VAN** or to **require the developer to**
 18744 provide coverage also for this case.
- 18745 **17.7.3.9** Similar considerations as for parameters hold for error messages specified in the
 18746 functional specification: Each error message, which belongs to a qualitatively distinct error case,
 18747 needs to be covered by testing. Note, that there may be exceptions, for example error messages
 18748 for errors, which cannot be provoked during testing. For such error messages other ways of
 18749 coverage need to be found as discussed in 15.2.2, "**Testing vs. alternate approaches to verify the**
 18750 **expected behaviour of functionality**".
- 18751 **17.7.3.9** Note that also the developer is allowed to use such alternative approaches to testing (e.
 18752 g. checking something in the source code) in the **coverage table**. **Of course the evaluator has to**
 18753 examine in this case, if this use of an alternative approach is acceptable (usually only in cases
 18754 where testing is practically impossible).
- 18755 **17.7.3.9** Depth (ATE_DPT) criteria) then it would be appropriate for the evaluator to give greater
 18756 focus to the repetition of developer tests. Note that this does not necessarily imply a requirement
 18757 for a high percentage sample for repetition of developer tests; indeed, given an extensive developer
 18758 test set, the evaluator may be able to justify a low percentage sample.
- 18759 Where the developer has used an automated test suite to perform functional testing, it will usually
 18760 be easier for the evaluator to re-run the entire test suite rather than repeat only a sample of
 18761 developer tests. However the evaluator does have an obligation to check that the automatic testing
 18762 does not give misrepresentative results. The implication is thus that this check must be performed
 18763 for a sample of the automatic test suite, with the principles for selecting some tests in preference to
 18764 others and ensuring a sufficient sample size applying equally in this case.
- 18765 The following principles should be followed whenever sampling is performed:
- 18766 a) Sampling should not be random, rather it should be chosen such that it is representative
 18767 of all of the evidence. The sample size and composition must always be justified.
- 18768 b) When sampling relates to the correct implementation of the TOE, the sample should be
 18769 representative of all aspects relevant to the areas that are sampled. In particular, the
 18770 selection should cover a variety of components, interfaces, developer and operational
 18771 sites (if more than one is involved) and hardware platform types (if more than one is
 18772 involved). The sample size should be commensurate with the cost effectiveness of the
 18773 evaluation and will depend on a number of TOE dependent factors (e.g. the size and
 18774 complexity of the TOE, the amount of documentation).
- 18775 c) Also, when sampling relates to specifically gaining evidence that the developer testing is
 18776 repeatable and reproducible the sample used must be sufficient to represent all distinct
 18777 aspects of developer testing, such as different test regimes. The sample used must be
 18778 sufficient to detect any systematic problem in the developer's functional testing process.
 18779 The evaluator contribution resulting from the combination of repeating developer tests
 18780 and performing independent tests must be sufficient to address the major points of
 18781 concern for the TOE.
- 18782 d) Where sampling relates to gaining evidence that a process (e.g. visitor control or design
 18783 review) the evaluator should sample sufficient information to gain reasonable confidence
 18784 that the procedure is being followed.
- 18785 e) The sponsor and developer should not be informed in advance of the exact composition of
 18786 the sample, subject to ensuring timely delivery of the sample and supporting deliverable,
 18787 e.g. test harnesses and equipment to the evaluator in accordance with the evaluation
 18788 schedule.

18789 f) The choice of the sample should be free from bias to the degree possible (one should not
18790 always choose the first or last item). Ideally the sample selection should be done by
18791 someone other than the evaluator.

18792 Errors found in the sample can be categorised as being either systematic or sporadic. If the error is
18793 systematic, the problem should be corrected and a complete new sample taken. If properly
18794 explained, sporadic errors might be solved without the need for a new sample, although the
18795 explanation should be confirmed. The evaluator should use judgement in determining whether to
18796 increase the sample size or use a different sample.

18797 **A.3 Dependencies**

18798 In general it is possible to perform the required evaluation activities, sub-activities, and actions in
18799 any order or in parallel. However, there are different kinds of dependencies which have to be
18800 considered by the evaluator. This Subclause provides general guidance on dependencies between
18801 different activities, sub-activities, and actions.

18802 **A.3.1 Dependencies between activities**

18803 For some cases the different assurance classes may recommend or even require a sequence for the
18804 related activities. A specific instance is the ST activity. The ST evaluation activity is started prior to
18805 any TOE evaluation activities since the ST provides the basis and context to perform them.
18806 However, a final verdict on the ST evaluation may not be possible until the TOE evaluation is
18807 complete, since changes to the ST may result from activity findings during the TOE evaluation.

18808 **A.3.2 Dependencies between sub-activities**

18809 Dependencies identified between components in ISO/IEC 15408-3 have to be considered by the
18810 evaluator. Most dependencies are one way, e.g. Evaluation of sub-activity (AVA_VAN.1) claims a
18811 dependency on Evaluation of sub-activity (ADV_FSP.1) and Evaluation of sub-activity (AGD_OPE.1).
18812 There are also instances of mutual dependencies, where both components depend on each other.
18813 An example of this is Evaluation of sub-activity (ATE_FUN.1) and Evaluation of sub-activity
18814 (ATE_COV.1).

18815 A sub-activity can be assigned a pass verdict normally only if all those sub-activities are
18816 successfully completed on which it has a one-way dependency. For example, a pass verdict on
18817 Evaluation of sub-activity (AVA_VAN.1) can normally only be assigned if the sub-activities related
18818 to Evaluation of sub-activity (ADV_FSP.1) and Evaluation of sub-activity (AGD_OPE.1) are assigned
18819 a pass verdict too. In the case of mutual dependency the ordering of these components is down to
18820 the evaluator deciding which sub-activity to perform first. Note this indicates that pass verdicts can
18821 normally only be assigned once both sub-activities have been successful.

18822 So when determining whether a sub-activity will impact another sub-activity, the evaluator should
18823 consider whether this activity depends on potential evaluation results from any dependent sub-
18824 activities. Indeed, it may be the case that a dependent sub-activity will impact this sub-activity,
18825 requiring previously completed evaluator actions to be performed again.

18826 A significant dependency effect occurs in the case of evaluator-detected flaws. If a flaw is identified
18827 as a result of conducting one sub-activity, the assignment of a pass verdict to a dependent sub-
18828 activity may not be possible until all flaws related to the sub-activity upon which it depends are
18829 resolved.

18830 **A.3.3 Dependencies between actions**

18831 It may be the case, that results which are generated by the evaluator during one action are used for
18832 performing another action. For example, actions for completeness and consistency cannot be
18833 completed until the checks for content and presentation have been completed. This means for

18834 example that the evaluator is recommended to evaluate the PP/ST rationale after evaluating the
18835 constituent parts of the PP/ST.

18836 **A.4 Site Visits**

18837 **A.4.1 Introduction**

18838 The assurance class ALC includes requirements for

18839 a) the application of configuration management, ensuring that the integrity of the TOE is
18840 preserved;

18841 b) measures, procedures, and standards concerned with secure delivery of the TOE,
18842 ensuring that the security protection offered by the TOE is not compromised during the
18843 transfer to the user,

18844 c) security measures, used to protect the development environment.

18845 A development site visit is a useful means whereby the evaluator determines whether procedures
18846 are being followed in a manner consistent with that described in the documentation.

18847 Reasons for visiting sites include:

18848 a) to observe the use of the CM system as described in the CM plan;

18849 b) to observe the practical application of delivery procedures as described in the delivery
18850 documentation;

18851 c) to observe the application of security measures during development and maintenance of
18852 the TOE as described in the development security documentation.

18853 Specific and detailed information is given in work units for those activities where site visits are
18854 performed:

18855 a) CM capabilities (ALC_CMC).n with $n \geq 3$ (especially work unit ALC_CMC.3-10 =
18856 ALC_CMC.4-13 = ALC_CMC.5-19);

18857 b) Delivery (ALC_DEL) (especially work unit ALC_DEL.1-2);

18858 c) Development security (ALC_DVS) (especially work unit ALC_DVS.1-3 = ALC_DVS.2-4).

18859 **A.4.2 General Approach**

18860 During an evaluation, it is often necessary that the evaluator will meet the developer more than
18861 once and it is a question of good planning to combine the site visit with another meeting to reduce
18862 costs. For example, one might combine the site visits for configuration management, for the
18863 developer's security and for delivery. It may also be necessary to perform more than one site visit
18864 to the same site to allow the checking of all development phases. It should be considered that
18865 development could occur at multiple facilities within a single building, multiple buildings at the
18866 same site, or at multiple sites.

18867 The first site visit should be scheduled early during the evaluation. In the case of an evaluation
18868 which starts during the development phase of the TOE, this will allow corrective actions to be
18869 taken, if necessary. In the case of an evaluation which starts after the development of the TOE, an
18870 early site visit could allow corrective measures to be put in place if serious deficiencies in the
18871 applied procedures emerge. This avoids unnecessary evaluation effort.

18872 Interviews are also a useful means of determining whether the written procedures reflect what is
18873 done. In conducting such interviews, the evaluator aims to gain a deeper understanding of the
18874 analysed procedures at the development site, how they are used in practise and whether they are
18875 being applied as described in the provided evaluation evidence. Such interviews complement but
18876 do not replace the examination of evaluation evidence.

18877 As a first step preparing the site visits the evaluators should perform the evaluator work units
18878 concerning the assurance class ALC excluding the aspects describing the results of the site visit.
18879 Based on the information provided by the relevant developer documentation and the remaining
18880 open questions which were not answered by the documentation the evaluators compile a check list
18881 of the questions which are to be resolved by the site visits.

18882 The first version of the evaluation report concerning the ALC class and the check list serves as
18883 input for the consultation with the evaluation authority concerning the site visits.

18884 The check list serves as a guide line for the site visits, which questions are to be answered by
18885 inspection of the relevant measures, their application and results, and by interviews. Where
18886 appropriate, sampling is used for gaining the required level of confidence (see Subclause A.2).

18887 The results of the site visits are recorded and serve as input for the final version of the evaluation
18888 report concerning the assurance class ALC.

18889 Other approaches to gain confidence should be considered that provide an equivalent level of
18890 assurance (e.g. to analyse evaluation evidence). Any decision not to make a visit should be
18891 determined in consultation with the evaluation authority. Appropriate security criteria and a
18892 methodology should be based on other standards of the Information Security Management Systems
18893 area.

18894 **A.4.3 Orientation Guide for the Preparation of the Check List**

18895 In the following some keywords are provided, which topics should be checked during an audit.

18896 **A.4.3.1 Aspects of configuration management**

18897 Basic

18898 — Items of the configuration list, including TOE, source code, run time libraries, design
18899 documentation, development tools (ALC_CMC.3-8).

18900 — Tracking of design documentation, source code, user guidance to different versions of the TOE.

18901 — Integration of the configuration system in the design and development process, test planning,
18902 test analysis and quality management procedures.

18903 Test analysis

18904 — Tracking of test plans and results to specific configurations and versions of the TOE.

18905 Access control to development systems

18906 — Policies for access control and logging.

18907 — Policies for project specific assignment and changing of access rights.

18908 Clearance

18909 — Policies for clearance of the TOE and user guidance to the customer.

- 18910 — Policies for testing and approving of components and the TOE before deployment.
- 18911 **A.4.3.2 Aspects of development security**
- 18912 Infrastructure
- 18913 — Security measures for physical access control to the development site and rationale for the
18914 effectiveness of these measures.
- 18915 Organisational measures
- 18916 — Organisational structure of the company in respect of the security of the development
18917 environment.
- 18918 — Organisational separation between development, production, testing and quality assurance.
- 18919 Personal measures
- 18920 — Measures for education of the personnel in respect of development security.
- 18921 — Measures and legal agreements of non-disclosure of internal information.
- 18922 Access control
- 18923 — Assignment of secured objects (for instance TOE, source code, run time libraries, design
18924 documentation, development tools, user guidance) and security policies.
- 18925 — Policies and responsibilities concerning the access control and the handling of authentication
18926 information.
- 18927 — Policies for logging of any kind access to the development site and protection of the logging
18928 data.
- 18929 Input, processing and output of data
- 18930 — Security measures for protection of output and output devices (printer, plotter and displays).
- 18931 — Securing of local networks and communication connections.
- 18932 Storage, transfer and destruction of documents and data media.
- 18933 — Policies for handling of documents and data media.
- 18934 — Policies and responsibilities for destruction of sorted out documents and logging of these
18935 events.
- 18936 Data protection
- 18937 — Policies and responsibilities for data and information protection (e.g. for performing backups).
- 18938 Contingency plan
- 18939 — Practises in case of emergency and responsibilities.
- 18940 — Documentation of the contingency measures concerning access control.

18941 — Information of the personnel about applicable practises in extreme cases. protection (e.g. for
18942 performing backups).

18943 **A.4.4 Example of a checklist**

18944 The examples of checklists for site visits consist in tables for the preparation of an audit and for the
18945 presentation of the results of an audit.

18946 The checklist structure given in the following is preliminary. Dependent on the concrete contents
18947 of the new guideline, changes might become necessary.

18948 The checklist is divided into three subclauses according to the subjects indicated in the
18949 introduction (Subclause A.4.1).

18950 a) Configuration management system.

18951 b) Delivery procedures.

18952 c) Security measures during development.

18953 These subclauses correspond to the actual ISO/IEC 15408 class ALC, especially the families CM
18954 capabilities (ALC_CMC).n with n>=3, Delivery (ALC_DEL) and Development security (ALC_DVS).

18955 The subclauses are subdivided further into rows corresponding to the relevant work units of this
18956 International Standard.

18957 The columns of the checklist contain in turn

18958 — a consecutive number,

18959 — the referenced work unit,

18960 — the references to the corresponding developer documentation,

18961 — the explicit reproduction of the developer measures,

18962 — special remarks and questions to be clarified on the visit (beyond the standard evaluator task
18963 to verify the application of the indicated measures),

18964 — the result of the examinations during the visit.

18965 If it is decided to have separate checklists for preparation and reporting of the audit, the result
18966 column is omitted in the preparation list and the remarks and questions column is omitted in the
18967 reporting list. The remaining columns should be identical in both lists.

18968 **Table A.1 Example of a checklist at EAL 4 (extract)**

A. Examination of the CM system (ALC_CMC.4 and ALC_CMS.4)					
No.	Work Unit	Developer Documentation	Measures	Questions and Remarks	Result
A.1	ALC_CMC.4-11, ALC_CMC.4-12	"Configuration Management System", ch. ...	The system automatically managing the source code files is capable of administering user profiles	Does reading or updating of a source code file require a user authentication?	If a user has not the right to access a confidential document, it is not even

A. Examination of the CM system (ALC_CMC.4 and ALC_CMS.4)					
No.	Work Unit	Developer Documentation	Measures	Questions and Remarks	Result
			and graded access rights, and of checking identification and authentication of users.		displayed to him in the file list.
...

B. Examination of the Delivery Procedures (ALC_DEL.1)					
No.	Work Unit	Developer Documentation	Measures	Questions and Remarks	Result
B.1	ALC_DEL.1-1, ALC_DEL.1-2	"Delivery of the TOE", ch. ...	The software is transmitted PGP-signed and encrypted to the customer.	---	The evaluators have checked the process and found it as described, additionally a checksum is transmitted.
...

C. Examination of the organisational and infrastructural developer security (ALC_DVS.1, ALC_LCD.1, ALC_TAT.1)					
No.	Work Unit	Developer Documentation	Measures	Questions and Remarks	Result
C.1	ALC_DVS.1-1, ALC_DVS.1-2	"Security of the development environment", ch. ... (Premises)	The premises are protected by security fencing.	Is the fencing sufficiently strong and high to prevent an easy intrusion into the premises?	The evaluators considered the fencing to be sufficiently strong and high.
C.2	ALC_DVS.1-1, ALC_DVS.1-2	"Security of the development environment", ch. ... (Building)	The building has the following access possibilities: The main entrance which is surveyed by the reception and is closed if the reception is	Is the listing of the access possibilities complete?	Beyond the indicated access possibilities, there is an emergency exit that cannot be opened from the outside. The roller shutters mentioned

C. Examination of the organisational and infrastructural developer security (ALC_DVS.1, ALC_LCD.1, ALC_TAT.1)					
No.	Work Unit	Developer Documentation	Measures	Questions and Remarks	Result
			not manned. And an access in the goods reception which is secured by two roller shutters.		before can be operated only from inside.
...

A.5 Scheme Responsibilities

This International Standard describes the minimum technical work that evaluations conducted under oversight (scheme) bodies must perform. However, it also recognises (both explicitly and implicitly) that there are activities or methods upon which mutual recognition of evaluation results do not rely. For the purposes of thoroughness and clarity, and to better delineate where this International Standard ends and an individual scheme's methodology begins, the following matters are left up to the discretion of the schemes. Schemes may choose to provide the following, although they may choose to leave some unspecified. (Every effort has been made to ensure this list is complete; evaluators encountering a subject neither listed here nor addressed in this International Standard should consult with their evaluation schemes to determine under whose auspices the subject falls.)

The matters that schemes may choose to specify include:

- a) what is required in ensuring that an evaluation was done sufficiently - every scheme has a means of verifying the technical competence, understanding of work and the work of its evaluators, whether by requiring the evaluators to present their findings to the oversight body, by requiring the oversight body to redo the evaluator's work, or by some other means that assures the scheme that all evaluation bodies are adequate and comparable;
- b) process for disposing of evaluation evidence upon completion of an evaluation;
- c) any requirements for confidentiality (on the part of the evaluator and the non-disclosure of information obtained during evaluation);
- d) the course of action to be taken if a problem is encountered during the evaluation (whether the evaluation continues once the problem is remedied, or the evaluation ends immediately and the remedied product must be re-submitted for evaluation);
- e) any specific (natural) language in which documentation must be provided;
- f) any recorded evidence that must be submitted in the ETR - this International Standard specifies the minimum to be reported in an ETR; however, individual schemes may require additional information to be included;
- g) any additional reports (other than the ETR) required from the evaluators -for example, testing reports;
- h) any specific ORs that may be required by the scheme, including the structure, recipients, etc. of any such ORs;

- 19002 i) any specific content structure of any written report as a result from an ST evaluation - a
19003 scheme may have a specific format for all of its reports detailing results of an evaluation,
19004 be it the evaluation of a TOE or of an ST;
- 19005 j) any additional PP/ST identification information required;
- 19006 k) any activities to determine the suitability of explicitly-stated requirements in an ST;
- 19007 l) any requirements for provision of evaluator evidence to support re-evaluation and re-use
19008 of evidence;
- 19009 m) any specific handling of scheme identifiers, logos, trademarks, etc.;
- 19010 n) any specific guidance in dealing with cryptography;
- 19011 o) handling and application of scheme, national and international interpretations;
- 19012 p) a list or characterisations of suitable alternative approaches to testing where testing is
19013 infeasible;
- 19014 q) the mechanism by which an evaluation authority can determine what steps an evaluator
19015 took while testing;
- 19016 r) preferred test approach (if any): at internal interface or at external interface;
- 19017 s) a list or characterisation of acceptable means of conducting the evaluator's vulnerability
19018 analysis (e.g. flaw hypothesis methodology);
- 19019 t) information regarding any vulnerabilities and weaknesses to be considered.

Annex B (informative)

Vulnerability Assessment (AVA)

19024 This annex provides an explanation of the AVA_VAN criteria and examples of their application. This
19025 annex does not define the AVA criteria; this definition can be found in ISO/IEC 15408-3 Subclause
19026 Class AVA: Vulnerability assessment.

19027 This annex consists of 2 major parts:

19028 a) *Guidance for completing an independent vulnerability analysis.* This is summarised in
19029 subclause B.1, and described in more detail in subclause B.2 . These subclauses describe
19030 how an evaluator should approach the construction of an independent Vulnerability
19031 Analysis.

19032 b) How to characterise and use assumed Attack Potential of an attacker. This is described in
19033 subclauses B.3 to B.5. These subclauses provide an example of describe how an attack
19034 potential can be characterised and should be used, and provide examples.

19035 **B.1 What is Vulnerability Analysis**

19036 The purpose of the vulnerability assessment activity is to determine the existence and
19037 exploitability of flaws or weaknesses in the TOE in the operational environment. This
19038 determination is based upon analysis performed by the evaluator, and is supported by evaluator
19039 testing.

19040 At the lowest levels of Vulnerability analysis (AVA_VAN) the evaluator simply performs a search of
19041 publicly available information to identify any known weaknesses in the TOE, while at the higher
19042 levels the evaluator performs a structured analysis of the TOE evaluation evidence.

19043 There are three main factors in performing a vulnerability analysis, namely:

19044 a) the identification of potential vulnerabilities;

19045 b) assessment to determine whether the identified potential vulnerabilities could allow an
19046 attacker with the relevant attack potential to violate the SFRs.

19047 c) penetration testing to determine whether the identified potential vulnerabilities are
19048 exploitable in the operational environment of the TOE.

19049 The identification of vulnerabilities can be further decomposed into the evidence to be searched
19050 and how hard to search that evidence to identify potential vulnerabilities. In a similar manner, the
19051 penetration testing can be further decomposed into analysis of the potential vulnerability to
19052 identify attack methods and the demonstration of the attack methods.

19053 These main factors are iterative in nature, i.e. penetration testing of potential vulnerabilities may
19054 lead to the identification of further potential vulnerabilities. Hence, these are performed as a single
19055 vulnerability analysis activity.

19056 **B.2 Evaluator construction of a Vulnerability Analysis**

19057 The evaluator vulnerability analysis is to determine that the TOE is resistant to penetration attacks
19058 performed by an attacker possessing a Basic (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for

19059 AVA_VAN.3), Moderate (for AVA_VAN.4) or High (for AVA_VAN.5) attack potential. The evaluator
 19060 first assesses the exploitability of all identified potential vulnerabilities. This is accomplished by
 19061 conducting penetration testing. The evaluator should assume the role of an attacker with a Basic
 19062 (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for AVA_VAN.3), Moderate (for AVA_VAN.4) or
 19063 High (for AVA_VAN.5) attack potential when attempting to penetrate the TOE.

19064 The evaluator considers potential vulnerabilities encountered by the evaluator during the conduct
 19065 of other evaluation activities. The evaluator penetration testing determining TOE resistance to
 19066 these potential vulnerabilities should be performed assuming the role of an attacker with a Basic
 19067 (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for AVA_VAN.3), Moderate (for AVA_VAN.4) or
 19068 High (for AVA_VAN.5) attack potential.

19069 However, vulnerability analysis should not be performed as an isolated activity. It is closely linked
 19070 with ADV and AGD. The evaluator performs these other evaluation activities with a focus on
 19071 identifying potential vulnerabilities or “areas of concern”. Therefore, evaluator familiarity with the
 19072 generic vulnerability guidance (provided in Subclause B.2.1) is required.

19073 **B.2.1 Generic vulnerability guidance**

19074 The following five categories provide discussion of generic vulnerabilities.

19075 **B.2.1.1 Bypassing**

19076 Bypassing includes any means by which an attacker could avoid security enforcement, by:

- 19077 a) exploiting the capabilities of interfaces to the TOE, or of utilities which can interact with
 19078 the TOE;
- 19079 b) inheriting privileges or other capabilities that should otherwise be denied;
- 19080 c) (where confidentiality is a concern) reading sensitive data stored or copied to
 19081 inadequately protected areas.

19082 Each of the following should be considered (where relevant) in the evaluator’s independent
 19083 vulnerability analysis.

- 19084 a) Attacks based on exploiting the capabilities of interfaces or utilities generally take
 19085 advantage of the absence of the required security enforcement on those interfaces. For
 19086 example, gaining access to functionality that is implemented at a lower level than that at
 19087 which access control is enforced. Relevant items include:

- 19088 1) changing the predefined sequence of invocation of TSFI;
- 19089 2) invoking an additional TSFI;
- 19090 3) using a component in an unexpected context or for an unexpected purpose;
- 19091 4) using implementation detail introduced in less abstract representations;
- 19092 5) using the delay between time of access check and time of use.

- 19093 b) Changing the predefined sequence of invocation of components should be considered
 19094 where there is an expected order in which interfaces to the TOE (e.g. user commands) are
 19095 called to invoke a TSFI (e.g. opening a file for access and then reading data from it). If a
 19096 TSFI is invoked through one of the TOE interfaces (e.g. an access control check), the
 19097 evaluator should consider whether it is possible to bypass the control by performing the
 19098 call at a later point in the sequence or by missing it out altogether.

- 19099 c) Executing an additional component (in the predefined sequence) is a similar form of
 19100 attack to the one described above, but involves the calling of some other TOE interface at
 19101 some point in the sequence. It can also involve attacks based on interception of sensitive
 19102 data passed over a network by use of network traffic analysers (the additional
 19103 component here being the network traffic analyser).
- 19104 d) Using a component in an unexpected context or for an unexpected purpose includes using
 19105 an unrelated TOE interface to bypass the TSF by using it to achieve a purpose that it was
 19106 not designed or intended to achieve. Covert channels are an example of this type of attack
 19107 (see B.2.1.4 for further discussion of covert channels). The use of undocumented
 19108 interfaces, which may be insecure, also falls into this category. Such interfaces may
 19109 include undocumented support and help facilities.
- 19110 e) Using implementation detail introduced in lower representations may allow an attacker
 19111 to take advantage of additional functions, resources or attributes that are introduced to
 19112 the TOE as a consequence of the refinement process. Additional functionality may include
 19113 test harness code contained in software modules and back-doors introduced during the
 19114 implementation process.
- 19115 f) Using the delay between time of check and time of use includes scenarios where an access
 19116 control check is made and access granted, and an attacker is subsequently able to create
 19117 conditions in which, had they applied at the time the access check was made, would have
 19118 caused the check to fail. An example would be a user creating a background process to
 19119 read and send highly sensitive data to the user's terminal, and then logging out and
 19120 logging back in again at a lower sensitivity level. If the background process is not
 19121 terminated when the user logs off, the MAC checks would have been effectively bypassed.
- 19122 g) Attacks based on inheriting privileges are generally based on illicitly acquiring the
 19123 privileges or capabilities of some privileged component, usually by exiting from it in an
 19124 uncontrolled or unexpected manner. Relevant items include:
- 19125 1) executing data not intended to be executable, or making it executable;
- 19126 2) generating unexpected input for a component;
- 19127 3) invalidating assumptions and properties on which lower-level components rely.
- 19128 h) Executing data not intended to be executable, or making it executable includes attacks
 19129 involving viruses (e.g. putting executable code or commands in a file which are
 19130 automatically executed when the file is edited or accessed, thus inheriting any privileges
 19131 the owner of the file has).
- 19132 i) Generating unexpected input for a component can have unexpected effects which an
 19133 attacker could take advantage of. For example, if the TSF could be bypassed if a user gains
 19134 access to the underlying operating system, it may be possible to gain such access
 19135 following the login sequence by exploring the effect of hitting various control or escape
 19136 sequences whilst a password is being authenticated.
- 19137 j) Invalidating assumptions and properties on which lower level components rely includes
 19138 attacks based on breaking out of the constraints of an application to gain access to an
 19139 underlying operating system in order to bypass the TSF of an application. In this case the
 19140 assumption being invalidated is that it is not possible for a user of the application to gain
 19141 such access. A similar attack can be envisaged against an application on an underlying
 19142 database management system: again the TSF could be bypassed if an attacker can break
 19143 out of the constraints of the application.

- 19144 k) Attacks based on reading sensitive data stored in inadequately protected areas
 19145 (applicable where confidentiality is a concern) include the following issues which should
 19146 be considered as possible means of gaining access to sensitive data:
- 19147 1) disk scavenging;
 - 19148 2) access to unprotected memory;
 - 19149 3) exploiting access to shared writable files or other shared resources (e.g. swap files);
 - 19150 4) Activating error recovery to determine what access users can obtain. For example, after a
 19151 crash an automatic file recovery system may employ a lost and found directory for
 19152 headerless files, which are on disk without labels. If the TOE implements mandatory
 19153 access controls, it is important to investigate at what security level this directory is kept
 19154 (e.g. at system high), and who has access to this directory.
- 19155 There are a number of different methods through which an evaluator may identify a back-door,
 19156 including two main techniques. Firstly, by the evaluator inadvertently identifying during testing an
 19157 interface that can be misused. Secondly, through testing each external interface of the TSF in a
 19158 debugging mode to identify any modules that are not called as a part of testing the documented
 19159 interfaces and then inspecting the code that is not called to consider whether it is a back-door.
- 19160 For a software TOE where Evaluation of sub-activity (ADV_IMP.2) and ALC_TAT.2 or higher
 19161 components are included in the assurance package, the evaluator may consider during their
 19162 analysis of the tools the libraries and packages that are linked by the compiler at compilation stage
 19163 to determine that back-doors are not introduced at this stage.
- 19164 **B.2.1.2 Tampering**
- 19165 Tampering includes any attack based on an attacker attempting to influence the behaviour of the
 19166 TSF (i.e. corruption or de-activation), for example by:
- 19167 a) accessing data on whose confidentiality or integrity the TSF relies;
 - 19168 b) forcing the TOE to cope with unusual or unexpected circumstances;
 - 19169 c) disabling or delaying security enforcement;
 - 19170 d) physical modification the TOE.
- 19171 Each of the following should be considered (where relevant) in the evaluator's independent
 19172 vulnerability analysis.
- 19173 a) Attacks based on accessing data, whose confidentiality or integrity are protected, include:
 - 19174 1) reading, writing or modifying internal data directly or indirectly;
 - 19175 2) using a component in an unexpected context or for an unexpected purpose;
 - 19176 3) using interfaces between components that are not visible at a higher level of abstraction.
 - 19177 b) Reading, writing or modifying internal data directly or indirectly includes the following
 19178 types of attack which should be considered:
 - 19179 1) reading "secrets" stored internally, such as user passwords;
 - 19180 2) spoofing internal data that security enforcing mechanisms rely upon;

- 19181 3) modifying environment variables (e.g. logical names), or data in configuration files or
19182 temporary files.
- 19183 c) It may be possible to deceive a trusted process into modifying a protected file that it
19184 wouldn't normally access.
- 19185 d) The evaluator should also consider the following "dangerous features":
- 19186 1) source code resident on the TOE along with a compiler (for instance, it may be possible to
19187 modify the login source code);
- 19188 2) an interactive debugger and patch facility (for instance, it may be possible to modify the
19189 executable image);
- 19190 3) the possibility of making changes at device controller level, where file protection does not
19191 exist;
- 19192 4) diagnostic code which exists in the source code and that may be optionally included;
- 19193 5) developer's tools left in the TOE.
- 19194 e) Using a component in an unexpected context or for an unexpected purpose includes (for
19195 example), where the TOE is an application built upon an operating system, users
19196 exploiting knowledge of a word processor package or other editor to modify their own
19197 command file (e.g. to acquire greater privileges).
- 19198 f) Using interfaces between components which are not visible at a higher level of
19199 abstraction includes attacks exploiting shared access to resources, where modification of
19200 a resource by one component can influence the behaviour of another (trusted)
19201 component, e.g. at source code level, through the use of global data or indirect
19202 mechanisms such as shared memory or semaphores.
- 19203 g) Attacks based on forcing the TOE to cope with unusual or unexpected circumstances
19204 should always be considered. Relevant items include:
- 19205 1) generating unexpected input for a component;
- 19206 2) invalidating assumptions and properties on which lower-level components rely.
- 19207 h) Generating unexpected input for a component includes investigating the behaviour of the
19208 TOE when:
- 19209 1) command input buffers overflow (possibly "crashing the stack" or overwriting other
19210 storage, which an attacker may be able to take advantage of, or forcing a crash dump that
19211 may contain sensitive information such as clear-text passwords);
- 19212 2) invalid commands or parameters are entered (including supplying a read-only parameter
19213 to an interface which expects to return data via that parameter and supplying improperly
19214 formatted input that should fail parsing such as SQL-injection, format strings);
- 19215 3) an end-of-file marker (e.g. CTRL-Z or CTRL-D) or null character is inserted in an audit
19216 trail.
- 19217 i) Invalidating assumptions and properties on which lower-level components rely includes
19218 attacks taking advantage of errors in the source code where the code assumes (explicitly
19219 or implicitly) that security relevant data is in a particular format or has a particular range
19220 of values. In these cases the evaluator should determine whether they can invalidate such

- 19221 assumptions by causing the data to be in a different format or to have different values,
19222 and if so whether this could confer advantage to an attacker.
- 19223 j) The correct behaviour of the TSF may be dependent on assumptions that are invalidated
19224 under extreme circumstances where resource limits are reached or parameters reach
19225 their maximum value. The evaluator should consider (where practical) the behaviour of
19226 the TOE when these limits are reached, for example:
- 19227 1) changing dates (e.g. examining how the TOE behaves when a critical date threshold is
19228 passed);
- 19229 2) filling disks;
- 19230 3) exceeding the maximum number of users;
- 19231 4) filling the audit log;
- 19232 5) saturating security alarm queues at a console;
- 19233 6) overloading various parts of a multi-user TOE which relies heavily upon communications
19234 components;
- 19235 7) swamping a network, or individual hosts, with traffic;
- 19236 8) filling buffers or fields.
- 19237 k) Attacks based on disabling or delaying security enforcement include the following items:
- 19238 1) using interrupts or scheduling functions to disrupt sequencing;
- 19239 2) disrupting concurrence;
- 19240 3) using interfaces between components which are not visible at a higher level of
19241 abstraction.
- 19242 l) Using interrupts or scheduling functions to disrupt sequencing includes investigating the
19243 behaviour of the TOE when:
- 19244 1) a command is interrupted (with CTRL-C, CTRL-Y, etc.);
- 19245 2) a second interrupt is issued before the first is acknowledged.
- 19246 m) The effects of terminating security critical processes (e.g. an audit daemon) should be
19247 explored. Similarly, it may be possible to delay the logging of audit records or the issuing
19248 or receipt of alarms such that it is of no use to an administrator (since the attack may
19249 already have succeeded).
- 19250 n) Disrupting concurrence includes investigating the behaviour of the TOE when two or
19251 more subjects attempt simultaneous access. It may be that the TOE can cope with the
19252 interlocking required when two subjects attempt simultaneous access, but that the
19253 behaviour becomes less well defined in the presence of further subjects. For example, a
19254 critical security process could be put into a resource-wait state if two other processes are
19255 accessing a resource which it requires.
- 19256 o) Using interfaces between components which are not visible at a higher level of
19257 abstraction may provide a means of delaying a time-critical trusted process.

- 19258 p) Physical attacks can be categorised into physical probing, physical manipulation, physical
19259 modification, and substitution.
- 19260 1) Physical probing by penetrating the TOE targeting internals of the TOE, e.g. reading at
19261 internal communication interfaces, lines or memories.
- 19262 2) Physical manipulation can be with the TOE internals aiming at internal modifications of
19263 the TOE (e.g. by using optical fault induction as an interaction process), at the external
19264 interfaces of the TOE (e.g. by power or clock glitches) and at the TOE environment (e.g. by
19265 modifying temperature).
- 19266 3) Physical modification of TOE internal security enforcing attributes to inherit privileges or
19267 other capabilities that should be denied in regular operation. Such modifications can be
19268 caused, e.g., by optical fault induction. Attacks based on physical modification may also
19269 yield a modification of the TSF itself, e.g. by causing faults at TOE internal program data
19270 transfers before execution. Note, that such kind of bypassing by modifying the TSF itself
19271 can jeopardise every TSF unless there are other measures (possibly environmental
19272 measures) that prevent an attacker from gaining physical access to the TOE.
- 19273 4) Physical substitution to replace the TOE with another IT entity, during delivery or
19274 operation of the TOE. Substitution during delivery of the TOE from the development
19275 environment to the user should be prevented through application of secure delivery
19276 procedures (such as those considered under Development security (ALC_DVS)).
19277 Substitution of the TOE during operation may be considered through a combination of
19278 user guidance and the operational environment, such that the user is able to be confident
19279 that they are interacting with the TOE.

19280 **B.2.1.3 Direct attacks**

19281 Direct attack includes the identification of any penetration tests necessary to test the strength of
19282 permutational or probabilistic mechanism and other mechanisms to ensure they withstand direct
19283 attack.

19284 For example, it may be a flawed assumption that a particular implementation of a pseudo-random
19285 number generator will possess the required entropy necessary to seed the security mechanism.

19286 Where a probabilistic or permutational mechanism relies on selection of security attribute value
19287 (e.g. selection of password length) or entry of data by a human user (e.g. choice of password), the
19288 assumptions made should reflect the worst case.

19289 Probabilistic or permutational mechanisms should be identified during examination of evaluation
19290 evidence required as input to this sub-activity (security target, functional specification, TOE design
19291 and implementation representation subset) and any other TOE (e.g. guidance) documentation may
19292 identify additional probabilistic or permutational mechanisms.

19293 Where the design evidence or guidance includes assertions or assumptions (e.g. about how many
19294 authentication attempts are possible per minute), the evaluator should independently confirm that
19295 these are correct. This may be achieved through testing or through independent analysis.

19296 Direct attacks reliant upon a weakness in a cryptographic algorithm should not be considered
19297 under Vulnerability analysis (AVA_VAN), as this is outside the scope of ISO/IEC 15408. Correctness
19298 of the implementation of the cryptographic algorithm is considered during the ADV and ATE
19299 activities.

19300 **B.2.1.4 Monitoring**

19301 Information is an abstract view on relation between the properties of entities, i.e. a signal contains
 19302 information for a system, if the TOE is able to react to this signal. The TOE resources processes and
 19303 stores information represented by user data. Therefore:

19304 a) information may flow with the user data between subjects by internal TOE transfer or
 19305 export from the TOE;

19306 b) information may be generated and passed to other user data;

19307 c) information may be gained through monitoring the operations on data representing the
 19308 information.

19309 The information represented by user data may be characterised by security attributes like
 19310 "classification level" having values, for example unclassified, confidential, secret, top secret, to
 19311 control operations to the data. This information and therefore the security attributes may be
 19312 changed by operations e.g. FDP_ACC.2 may describe decrease of the level by "sanitarisation" or
 19313 increase of level by combination of data. This is one aspects of an information flow analysis focused
 19314 on controlled operations of controlled subjects on controlled objects.

19315 The other aspect is the analysis of *illicit information flow*. This aspect is more general than the
 19316 direct access to objects containing user data addressed by the FDP_ACC family. An *unenforced*
 19317 signalling channel carrying information under control of the information flow control policy can
 19318 also be caused by monitoring of the processing of any object containing or related to this
 19319 information (e.g. side channels). An *enforced* signalling channels may be identified in terms of the
 19320 subjects manipulating resources and the subject or user that observe such manipulation.
 19321 Classically, covert channels have been identified as timing or storage channels, according to the
 19322 resource being modified or modulated. As for other monitoring attacks, the use of the TOE is in
 19323 accordance with the SFRs.

19324 Covert channels are normally applicable in the case when the TOE has unobservability AND multi-
 19325 level separation policy requirements. Covert channels may be routinely spotted during
 19326 vulnerability analysis and design activities, and should therefore be tested. However, generally
 19327 such monitoring attacks are only identified through specialised analysis techniques commonly
 19328 referred to as "covert channel analysis". These techniques have been the subject of much research
 19329 and there are many papers published on this subject. Guidance for the conduct of covert channel
 19330 analysis should be sought from the evaluation authority.

19331 *Unenforced* information flow monitoring attacks include passive analysis techniques aiming at
 19332 disclosure of sensitive internal data of the TOE by operating the TOE in the way that corresponds
 19333 to the guidance documents.

19334 Side Channel Analysis includes crypt analytical techniques based on physical leakage of the TOE.
 19335 Physical leakage can occur by timing information, power consumption or power emanation during
 19336 computation of a TSF. Timing information can be collected also by a remote-attacker (having
 19337 network access to the TOE), power based information channels requires that the attacker is in the
 19338 near-by environment of the TOE.

19339 Eavesdropping techniques include interception of all forms of energy, e.g., electromagnetic or
 19340 optical emanation of computer displays, not necessarily in the near-field of the TOE.

19341 Monitoring also includes exploits of protocol flaws, e.g., an attack on SSL implementation.

19342 **B.2.1.5 Misuse**

19343 Misuse may arise from:

- 19344 a) incomplete guidance documentation;
- 19345 b) unreasonable guidance;
- 19346 c) unintended misconfiguration of the TOE;
- 19347 d) forced exception behaviour of the TOE.
- 19348 If the guidance documentation is incomplete the user may not know how to operate the TOE in
 19349 accordance with the SFRs. The evaluator should apply familiarity with the TOE gained from
 19350 performing other evaluation activities to determine that the guidance is complete. In particular, the
 19351 evaluator should consider the functional specification. The TSF described in this document should
 19352 be described in the guidance as required to permit secure administration and use through the TSFI
 19353 available to human users. In addition, the different modes of operation should be considered to
 19354 ensure that guidance is provided for all modes of operation.
- 19355 The evaluator may, as an aid, prepare an informal mapping between the guidance and these
 19356 documents. Any omissions in this mapping may indicate incompleteness.
- 19357 The guidance is considered to be unreasonable if it makes demands on the TOE's usage or
 19358 operational environment that are inconsistent with the ST or unduly onerous to maintain security.
- 19359 A TOE may use a variety of ways to assist the consumer in effectively using that TOE in accordance
 19360 with the SFRs and prevent unintentional misconfiguration. A TOE may employ functionality
 19361 (features) to alert the consumer when the TOE is in a state that is inconsistent with the SFRs, whilst
 19362 other TOEs may be delivered with enhanced guidance containing suggestions, hints, procedures,
 19363 etc. on using the existing security features most effectively; for instance, guidance on using the
 19364 audit feature as an aid for detecting when the SFRs are being compromised; namely insecure.
- 19365 The evaluator considers the TOE's functionality, its purpose and security objectives for the
 19366 operational environment to arrive at a conclusion of whether or not there is reasonable
 19367 expectation that use of the guidance would permit transition into an insecure state to be detected
 19368 in a timely manner.
- 19369 **17.7.3.12 The potential for the TOE to enter into insecure states may be determined using**
 19370 **the evaluation deliverables, such as the ST, the functional specification and any other**
 19371 **design representations provided as evidence for components included in the assurance**
 19372 **package for the TOE (e.g. the TOE/TSF design specification if a component from Objectives**
- 19373 **17.7.3.12** The objectives of this sub-activity are to determine whether the formal security policy model of
 19374 the TSF clearly and consistently describes the rules and characteristics of the security policies
 19375 and whether this description corresponds with the description of security functions in the
 19376 functional specification.
- 19377 **17.7.3.12 Input**
- 19378 **17.7.3.12** The evaluation evidence for this sub-activity is:
- 19379 **17.7.3.12** the ST;
- 19380 **17.7.3.12** the functional specification;
- 19381 **17.7.3.12** formal security policy model (ADV_SPM.1.1D);
- 19382 **17.7.3.12** formal proof of correspondence between the model and any formal functional specification
 19383 (ADV_SPM.1.3D);

19384	17.7.3.12 demonstration of correspondence between the model and the functional specification
19385	(ADV_SPM.1.4D).
19386	17.7.3.12 Application notes
19387	17.7.3.12 This activity applies to cases where the developer has provided a formal security
19388	policy model of the TOE.
19389	17.7.3.12 A formal TOE security policy model is a representation of the rules (synonymously
19390	termed “principles”) of security policies and characteristics of the TSF behaviour in mathematical
19391	terms. Their formal counterparts are called security properties and security features,
19392	respectively. The representation includes but is not limited to algebraic specifications, finite state
19393	machines and logic formalisms strong enough to formally infer the properties from the features.
19394	The formal TSP model is accompanied by an informal interpretation explaining how the rules and
19395	characteristics are mapped to the respective properties and features.
19396	17.7.3.12 The creation of a formal security policy model helps to identify and eliminate
19397	ambiguous, inconsistent, contradictory, or unenforceable security policy elements. Once the TOE
19398	has been built, the formal model serves the evaluation effort by contributing to the evaluator's
19399	judgement of how well the developer has understood the security functionality being
19400	implemented and whether there are inconsistencies between the security requirements and the
19401	TOE design. The confidence in the model is accompanied by a proof that it contains no
19402	inconsistencies.
19403	17.7.3.12 A formal security model is a precise formal presentation of the important aspects of
19404	security and their relationship to the behaviour of the TOE; it identifies the set of rules
19405	(principles) that defines the TOE security policy and the set of practises (characteristics) that
19406	regulates how the TSF manages, protects, and otherwise controls the system resources. The
19407	model includes the set of restrictions and properties that specify how information and computing
19408	resources are prevented from being used to violate the SFRs, accompanied by a persuasive set of
19409	engineering arguments showing that these restrictions and properties play a key role in the
19410	enforcement of the SFRs. It consists both of the formalisms that express the security functionality,
19411	as well as ancillary text to explain the model and to provide it with context. The security
19412	behaviour of the TSF is modelled both in terms of external behaviour (i.e. how the TSF interacts
19413	with the rest of the TOE and with its operational environment), as well as its internal behaviour.
19414	17.7.3.12 The Security Policy Model of the TOE is informally abstracted from its realisation by
19415	considering the proposed security requirements of the ST. The informal abstraction is taken to be
19416	successful if the TOE's principles turn out to be enforced by its characteristics. The purpose of
19417	formal methods lies within the enhancement of the rigour of enforcement. Informal arguments
19418	are always prone to fallacies; especially if relationships among subjects, objects and operations
19419	get more and more involved. In order to minimise the risk of insecure state arrivals the rules and
19420	characteristics of the security policy model are mapped to respective properties and features
19421	within some formal system, whose rigour and strength can afterwards be used to obtain the
19422	security properties by means of theorems and formal proof.
19423	17.7.3.12 While the term “formal security policy model” is used in academic circles, the CC's
19424	approach has no fixed definition of “security”; it would equate to whatever SFRs are being
19425	claimed. Therefore, the formal security policy model is merely a formal representation of the set
19426	of SFRs being claimed.
19427	17.7.3.12 The term security policy has traditionally been associated with only access control
19428	policies, whether label-based (mandatory access control) or user-based (discretionary access
19429	control). However, a security policy is not limited to access control; there are also audit policies,
19430	identification policies, authentication policies, encryption policies, management policies, and any
19431	other security policies that are enforced by the TOE, as described in the PP/ST. ADV_SPM.1.1D
19432	contains an assignment for identifying these policies that are formally modelled.

- 19433 **17.7.3.12** It is recognized that not all policies can be formally modelled for all TOEs. This is
 19434 because either a given policy can not be formally modelled in the otherwise well suited
 19435 framework, or because the nature of the TOE renders impossible the modelling of policies that
 19436 would otherwise be possible to model.
- 19437 **17.7.3.12 Action ADV_SPM.1.1E**
- 19438 **17.7.3.12 ADV_SPM.1.1C** *The model shall be in a formal style, supported by explanatory*
 19439 *text as required, and identify the security policies of the TSF that are modelled.*
- 19440 **17.7.3.12 Work unit ADV_SPM.1-1**
- 19441 **17.7.3.12** The evaluator ***shall examine the TOE security policy model to determine that it is***
 19442 ***written in a formal style.***
- 19443 **17.7.3.12** The evaluator identifies the formal framework upon which the TOE security policy
 19444 model is based and ensures that it is founded on well established mathematical concepts. **They**
 19445 **also identify the security properties and features addressed in the application notes and ensure**
 19446 **the formalization of at least one security policy.**
- 19447 **17.7.3.12** For guidance on formal methods refer to ISO/IEC **15408-3**
- 19448 **17.7.3.12 Work unit ADV_SPM.1-2**
- 19449 **17.7.3.12** The evaluator ***shall examine the TOE security policy model to determine that it***
 19450 ***contains all necessary informal explanatory text.***
- 19451 **17.7.3.12** Supporting narrative descriptions are necessary for all parts of the model (for
 19452 example, to make clear the meaning of any formal notation and how they are used) including the
 19453 security properties and features.
- 19454 **17.7.3.12 Work unit ADV_SPM.1-3**
- 19455 **17.7.3.12** The evaluator ***shall examine the TOE security policy model to determine that all***
 19456 ***security policies of the TSF are identified that are modelled.***
- 19457 **17.7.3.12** The evaluator determines whether the SPM identifies the security policies for which a
 19458 model is provided, identifying the relevant portions of the statement of SFRs that comprise each
 19459 of the modelled policies.
- 19460 **17.7.3.12** The evaluator determines whether the list of security policies identified by the SPM is
 19461 consistent with the assignment of ADV_SPM.1.1D in the ST.
- 19462 **17.7.3.12** The evaluator determines whether for each security policy identified by the SPM a
 19463 model is in fact provided.
- 19464 **17.7.3.12 ADV_SPM.1.2C** *For all policies that are modelled, the model shall define*
 19465 *security for the TOE and provide a formal proof that the TOE cannot reach a state that is*
 19466 *not secure.*
- 19467 **17.7.3.12 Work unit ADV_SPM.1-4**
- 19468 **17.7.3.12** The evaluator ***shall examine the principles and characteristics of the security policies***
 19469 ***to determine that the modelled security behaviour of the TOE is clearly articulated.***
- 19470 **17.7.3.12** The security policies are expressed in terms of security principles (rules) which are
 19471 modelled by security properties and define the secure state of the TOE. For example, a model

- 19472 based on state transitions could describe the security policies in terms of principles of its states,
19473 identify its initial state, and define what it means to be a secure state.
- 19474 **17.7.3.12** The evaluator determines that the security policies are reflected within their formal
19475 counterparts of the TSP model.
- 19476 **17.7.3.12** The TOE security behaviour is expressed in terms of security characteristics (i.e.
19477 portions of TOE security functionality managing, protecting, and otherwise controlling the system
19478 resources including attributes and conditions of the TOE) which are modelled by security
19479 features. For example, a model based on state transitions could describe the characteristics as
19480 possible actions in each secure state in a level of detail sufficient to decide into which state the
19481 TOE will be transformed by that action.
- 19482 **17.7.3.12** Together the security principles and characteristics describe the entire security
19483 posture of the TOE.
- 19484 **17.7.3.12** In the context of a formal TOE security policy model the security behaviour is
19485 considered to be clearly articulated only if an adequate mapping from principles and
19486 characteristics to their respective formal counterparts properties and features has been given.
19487 The mapping is considered to be adequate if the level of abstraction from the TOE's realization is
19488 detailed enough to allow for correct identification of all security objectives and the relation to the
19489 security environment.
- 19490 **17.7.3.12** The above condition for clear articulation is necessary but not sufficient. An informal
19491 interpretation of all formal concepts (including attributes, predicates and variables, if available)
19492 must be provided in order to make clear their intended meaning.
- 19493 **17.7.3.12 Work unit ADV_SPM.1-5**
- 19494 **17.7.3.12** The evaluator *shall examine the TOE security policy model rationale to determine*
19495 *that it formally proves that the security features enforce the security properties.*
- 19496 **17.7.3.12** To determine the enforcement, the evaluator considers the security properties and
19497 the security features and verifies that the arguments used in the proof are valid. The proof of
19498 correspondence between the security properties and the security features shall be formal.
- 19499 **17.7.3.12** The validity of the security properties shall mean that the TOE is in a secure state. By
19500 this, the evaluator confirms by means of the rationale that the TOE never reaches an insecure
19501 state.
- 19502 **17.7.3.12 Work unit ADV_SPM.1-6**
- 19503 **17.7.3.12** The evaluator *shall examine the TOE security policy model rationale to determine*
19504 *that it proves the internal consistency of the TOE security policy model.*
- 19505 **17.7.3.12** The proof shall show the absence of contradictions within the TOE security policy
19506 model. In determining the absence of contradictions, the evaluator verifies that the arguments
19507 used in the proof are valid.
- 19508 **17.7.3.12** Since the TOE security policy model is formal, the proof of its internal consistency
19509 shall be formal. It is recognized that a complete formal proof of the internal consistency of the
19510 TOE security policy model usually is not possible due to the fundamental nature of formal
19511 frameworks. Generally, it is sufficient to generate evidence using formal proofs based on the
19512 specific TOE security policy model that prove the internal consistency by means of a combination
19513 with generic arguments of the formal framework.
- 19514 **17.7.3.12 ADV_SPM.1.3C** *The correspondence between the model and the functional*
19515 *specification shall be at the correct level of formality.*

- 19516 **17.7.3.12 Work unit ADV_SPM.1-7**
- 19517 **17.7.3.12** The evaluator ***shall examine the correspondence between the model and the***
 19518 functional specification to determine that a semiformal demonstration of correspondence
 19519 between the model and any semiformal functional specification is provided.
- 19520 **17.7.3.12** This work unit is only applicable to a semiformal presentation of the functional
 19521 specification, which is required by ADV_FSP.5.2C.
- 19522 **17.7.3.12** A semiformal correspondence is one that results from a structured approach with a
 19523 substantial degree of rigor (in terms of completeness and correctness), but is not as rigorous as a
 19524 mathematical proof. Such a semiformal correspondence limits the subjective interpretations of its
 19525 terms, and so it provides less ambiguity than would exist in an informal correspondence.
- 19526 **17.7.3.12** For guidance on semiformal methods refer to Annex 3.1.1 '**Semiformal and formal**
 19527 methods'.
- 19528 **17.7.3.12 Work unit ADV_SPM.1-8**
- 19529 **17.7.3.12** The evaluator ***shall examine the correspondence between the model and the***
 19530 functional specification to determine that a formal proof of correspondence between the model
 19531 and any formal functional specification is provided.
- 19532 **17.7.3.12** This work unit is only applicable to a formal presentation of the functional
 19533 specification, which is required by ADV_FSP.6.2D.
- 19534 **17.7.3.12** There should be a formal proof of correspondence between the model and any formal
 19535 functional specification.
- 19536 **17.7.3.12** The formal proof of correspondence removes all subjective interpretations of its
 19537 terms by enlisting well-established mathematical concepts to define the syntax and semantics of
 19538 the formal notation and uses rules that support logical reasoning. The security features within
 19539 the TOE (which are identified in the formal TSP model) are expressed in a formal specification
 19540 language and shown to be satisfied by the formal specification.
- 19541 **17.7.3.12** For guidance on formal methods refer to **ISO/IEC 15408-3**.
- 19542 **17.7.3.12 ADV_SPM.1.4C** ***The correspondence shall show that the functional***
 19543 ***specification is consistent and complete with respect to the model.***
- 19544 **17.7.3.12 Work unit ADV_SPM.1-9**
- 19545 **17.7.3.12** The evaluator ***shall examine the correspondence to determine that the behaviour at***
 19546 the TSF interfaces (as articulated in the functional specification) is complete with respect to the
 19547 behaviour modelled by the security features.
- 19548 **17.7.3.12** The term "correspondence" here means both the formal proof of correspondence
 19549 between the formal SPM and any formal FSP required by ADV_SPM.1.2D and the demonstration
 19550 of correspondence between the formal SPM and the FSP required by ADV_SPM.1.3D.
- 19551 **17.7.3.12** In determining completeness of the correspondence, the evaluator considers the
 19552 description of TSFI behaviour and maps adequate portions (characteristics) to corresponding
 19553 features of the TSP model. The demonstration should show that all characteristics belonging to
 19554 policies that are required to be modelled have an associated feature description in the TOE
 19555 security policy model, and that each feature of the TSP model does occur in the mapping.
- 19556 **17.7.3.12** Abstention from formally modelling TSFI behaviour always calls for justification on
 19557 the developer's side (also confer the application notes above).

19558 **17.7.3.12 Work unit ADV_SPM.1-10**

19559 **17.7.3.12** The evaluator *shall examine the correspondence to determine that the behaviour at*
 19560 *the TSF interfaces (as articulated in the functional specification) is consistent with respect to the*
 19561 *behaviour modelled by the security features.*

19562 **17.7.3.12** The term “correspondence” here means both the formal proof of correspondence
 19563 between the formal SPM and any formal FSP required by ADV_SPM.1.3D and the demonstration
 19564 of correspondence between the SPM and the FSP required by ADV_SPM.1.4D.

19565 **17.7.3.12** The meaning of consistency reflects the conventional understanding in contrast to the
 19566 internal consistency concept of work unit ADV_SPM.1-6.

19567 **17.7.3.12** In determining consistency, the evaluator resumes the mapping of TSFI behaviour to
 19568 security features established in the preceding work unit and verifies that the correspondence
 19569 shows that each security feature of the TSP model accurately reflects the corresponding TSFI
 19570 behaviour.

19571 **17.7.3.12** For example, if TSFI behaviour dealt with access management on the granularity of
 19572 single individuals, then a TSP model describing the security behaviour of the TOE in terms of
 19573 groups of users would not be consistent. Likewise, if TSFI behaviour dealt with access
 19574 management for groups of users, then a TSP model describing the security behaviour of the TOE
 19575 in terms of individual users would also not be consistent.

19576 **17.7.3.12** As another example, if remote untrusted users had to pass more stringent
 19577 authentication procedures than administrators whose only point of access were within a
 19578 physically-protected area, then this difference in authentication procedures had to be reflected in
 19579 the security features.

19580 **17.7.3.12** TOE design (ADV_TDS) is included).

19581 Instances of forced exception behaviour of the TSF could include, but are not limited to, the
 19582 following:

- 19583 a) behaviour of the TOE when start-up, close-down or error recovery is activated;
- 19584 b) behaviour of the TOE under extreme circumstances (sometimes termed overload or
 19585 asymptotic behaviour), particularly where this could lead to the de-activation or
 19586 disabling of parts of the TSF;
- 19587 c) any potential for unintentional misconfiguration or insecure use arising from attacks
 19588 noted in the subclause on tampering above.

19589 **B.2.2 Identification of Potential Vulnerabilities**

19590 Potential vulnerabilities may be identified by the evaluator during different activities. They may
 19591 become apparent during an evaluation activity or they may be identified as a result of analysis of
 19592 evidence to search for vulnerabilities.

19593 **B.2.2.1 Encountered**

19594 The encountered identification of vulnerabilities is where potential vulnerabilities are identified by
 19595 the evaluator during the conduct of evaluation activities, i.e. the evidence are not being analysed
 19596 with the express aim of identifying potential vulnerabilities.

19597 The encountered method of identification is dependent on the evaluator's experience and
 19598 knowledge; which is monitored and controlled by the evaluation authority. It is not reproducible in

- 19599 approach, but will be documented to ensure repeatability of the conclusions from the reported
19600 potential vulnerabilities.
- 19601 There are no formal analysis criteria required for this method. Potential vulnerabilities are
19602 identified from the evidence provided as a result of knowledge and experience. However, this
19603 method of identification is not constrained to any particular subset of evidence.
- 19604 Evaluator is assumed to have knowledge of the TOE-type technology and known security flaws as
19605 documented in the public domain. The level of knowledge assumed is that which can be gained
19606 from a security e-mail list relevant to the TOE type, the regular bulletins (bug, vulnerability and
19607 security flaw lists) published by those organisations researching security issues in products and
19608 technologies in widespread use. This knowledge is not expected to extend to specific conference
19609 proceedings or detailed theses produced by university research for AVA_VAN.1 or AVA_VAN.2.
19610 However, to ensure the knowledge applied is up to date, the evaluator may need to perform a
19611 search of public domain material.
- 19612 For AVA_VAN.3 to AVA_VAN.5 the search of publicly available information is expected to include
19613 conference proceeding and theses produced during research activities by universities and other
19614 relevant organisations.
- 19615 Examples of how these may arise (how the evaluator may encounter potential vulnerabilities):
- 19616 a) while the evaluator is examining some evidence, it sparks a memory of a potential
19617 vulnerability identified in a similar product type, that the evaluator believes to also be
19618 present in the TOE under evaluation;
- 19619 b) while examining some evidence, the evaluator spots a flaw in the specification of an
19620 interface, that reflects a potential vulnerability.
- 19621 This may include becoming aware of a potential vulnerability in a TOE through reading about
19622 generic vulnerabilities in a particular product type in an IT security publication or on a security e-
19623 mail list to which the evaluator is subscribed.
- 19624 Attack methods can be developed directly from these potential vulnerabilities. Therefore, the
19625 encountered potential vulnerabilities are collated at the time of producing penetration tests based
19626 on the evaluator's vulnerability analysis. There is no explicit action for the evaluator to encounter
19627 potential vulnerabilities. Therefore, the evaluator is directed through an implicit action specified in
19628 AVA_VAN.1.2E and AVA_VAN.*4E.
- 19629 Current information regarding public domain vulnerabilities and attacks may be provided to the
19630 evaluator by, for example, an evaluation authority. This information is to be taken into account by
19631 the evaluator when collating encountered vulnerabilities and attack methods when developing
19632 penetration tests.
- 19633 **B.2.2.2 Analysis**
- 19634 The following types of analysis are presented in terms of the evaluator actions.
- 19635 **B.2.2.2.1 Unstructured Analysis**
- 19636 The unstructured analysis to be performed by the evaluator (for Evaluation of sub-activity
19637 (AVA_VAN.2)) permits the evaluator to consider the generic vulnerabilities (as discussed in B.2.1).
19638 The evaluator will also apply their experience and knowledge of flaws in similar technology types.
- 19639 **B.2.2.2.2 Focused**
- 19640 During the conduct of evaluation activities, the evaluator may also identify areas of concern. These
19641 are specific portions of the TOE evidence that the evaluator has some reservation about, although

- 19642 the evidence meets the requirements for the activity with which the evidence is associated. For
 19643 example, a particular interface specification looks particularly complex, and therefore may be
 19644 prone to error either in the development of the TOE or in the operation of the TOE. There is no
 19645 potential vulnerability apparent at this stage, further investigation is required. This is beyond the
 19646 bounds of encountered, as further investigation is required.
- 19647 Difference between potential vulnerability and area of concern:
- 19648 a) Potential vulnerability - The evaluator knows a method of attack that can be used to
 19649 exploit the weakness or the evaluator knows of vulnerability information that is relevant
 19650 to the TOE.
- 19651 b) Area of concern - The evaluator may be able to discount concern as a potential
 19652 vulnerability based on information provided elsewhere. While reading interface
 19653 specification, the evaluator identifies that due to the extreme (unnecessary) complexity
 19654 of an interface a potential vulnerability may lay within that area, although it is not
 19655 apparent through this initial examination.
- 19656 The focused approach to the identification of vulnerabilities is an analysis of the evidence with the
 19657 aim of identifying any potential vulnerabilities evident through the contained information. It is an
 19658 unstructured analysis, as the approach is not predetermined. This approach to the identification of
 19659 potential vulnerabilities can be used during the independent vulnerability analysis required by
 19660 Evaluation of sub-activity (AVA_VAN.3).
- 19661 This analysis can be achieved through different approaches, that will lead to commensurate levels
 19662 of confidence. None of the approaches have a rigid format for the examination of evidence to be
 19663 performed.
- 19664 The approach taken is directed by the results of the evaluator's assessment of the evidence to
 19665 determine it meets the requirements of the AVA/AGD sub-activities. Therefore, the investigation of
 19666 the evidence for the existence of potential vulnerabilities may be directed by any of the following:
- 19667 a) areas of concern identified during examination of the evidence during the conduct of
 19668 evaluation activities;
- 19669 b) reliance on particular functionality to provide separation, identified during the analysis of
 19670 the architectural design (as in Evaluation of sub-activity (ADV_ARC.1)), requiring further
 19671 analysis to determine it cannot be bypassed;
- 19672 c) representative examination of the evidence to hypothesise potential vulnerabilities in the
 19673 TOE.
- 19674 The evaluator will report what actions were taken to identify potential vulnerabilities in the
 19675 evidence. However, the evaluator may not be able to describe the steps in identifying potential
 19676 vulnerabilities before the outset of the examination. The approach will evolve as a result of the
 19677 outcome of evaluation activities.
- 19678 The areas of concern may arise from examination of any of the evidence provided to satisfy the
 19679 SARs specified for the TOE evaluation. The information publicly accessible is also considered.
- 19680 The activities performed by the evaluator can be repeated and the same conclusions, in terms of
 19681 the level of assurance in the TOE, can be reached although the steps taken to achieve those
 19682 conclusions may vary. As the evaluator is documenting the form the analysis took, the actual steps
 19683 taken to achieve those conclusions are also reproducible.

19684 **B.2.2.2.3 Methodical**

19685 The methodical analysis approach takes the form of a structured examination of the evidence. This
 19686 method requires the evaluator to specify the structure and form the analysis will take (i.e. the
 19687 manner in which the analysis is performed is predetermined, unlike the focused identification
 19688 method). The method is specified in terms of the information that will be considered and how/why
 19689 it will be considered. This approach to the identification of potential vulnerabilities can be used
 19690 during the independent vulnerability analysis required by Evaluation of sub-activity (AVA_VAN.4)
 19691 and Evaluation of sub-activity (AVA_VAN.5).

19692 This analysis of the evidence is deliberate and pre-planned in approach, considering all evidence
 19693 identified as an input into the analysis.

19694 All evidence provided to satisfy the (ADV) assurance requirements specified in the assurance
 19695 package are used as input to the potential vulnerability identification activity.

19696 The “methodical” descriptor for this analysis has been used in an attempt to capture the
 19697 characterisation that this identification of potential vulnerabilities is to take an ordered and
 19698 planned approach. A “method” or “system” is to be applied in the examination. The evaluator is to
 19699 describe the method to be used in terms of what evidence will be considered, the information
 19700 within the evidence that is to be examined, the manner in which this information is to be
 19701 considered; and the hypothesis that is to be generated.

19702 The following provide some examples that a hypothesis may take:

19703 a) consideration of malformed input for interfaces available to an attacker at the external
 19704 interfaces;

19705 b) examination of a security mechanism, such as domain separation, hypothesising internal
 19706 buffer overflows leading to degradation of separation;

19707 c) analysis to identify any objects created in the TOE implementation representation that
 19708 are then not fully controlled by the TSF, and could be used by an attacker to undermine
 19709 the SFRs.

19710 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE
 19711 and specify an approach to the analysis that “all interface specifications provided in the functional
 19712 specification and TOE design will be analysed to hypothesise potential vulnerabilities” and go on to
 19713 explain the methods used in the hypothesis.

19714 This identification method will provide a plan of attack of the TOE, that would be performed by an
 19715 evaluator completing penetration testing of potential vulnerabilities in the TOE. The rationale for
 19716 the method of identification would provide the evidence for the coverage and depth of exploitation
 19717 determination that would be performed on the TOE.

19718 **B.3 When attack potential is used**

19719 **B.3.1 Developer**

19720 Attack potential is used by a PP/ST author during the development of the PP/ST, in consideration
 19721 of the threat environment and the selection of assurance components. This may simply be a
 19722 determination that the attack potential possessed by the assumed attackers of the TOE is
 19723 generically characterised as Basic, Enhanced-Basic, Moderate or High. Alternatively, the PP/ST may
 19724 wish to specify particular levels of individual factors assumed to be possessed by attackers. (e.g.
 19725 the attackers are assumed to be experts in the TOE technology type, with access to specialised
 19726 equipment.)

19727 The PP/ST author considers the threat profile developed during a risk assessment (outside the
 19728 scope of ISO/IEC 15408, but used as an input into the development of the PP/ST in terms of the
 19729 Security Problem Definition or in the case of Direct Rationale STs, the requirements statement).
 19730 Consideration of this threat profile in terms of one of the approaches discussed in the following
 19731 subclauses will permit the specification of the attack potential the TOE is to resist.

19732 **B.3.2 Evaluator**

19733 Attack potential is especially considered by the evaluator in two distinct ways during the ST
 19734 evaluation and the vulnerability assessment activities.

19735 Attack potential is used by an evaluator during the conduct of the vulnerability analysis sub-
 19736 activity to determine whether or not the TOE is resistant to attacks assuming a specific attack
 19737 potential of an attacker. If the evaluator determines that a potential vulnerability is exploitable in
 19738 the TOE, they have to confirm that it is exploitable considering all aspects of the intended
 19739 environment, including the attack potential assumed by an attacker.

19740 Therefore, using the information provided in the threat statement of the Security Target, the
 19741 evaluator determines the minimum attack potential required by an attacker to effect an attack, and
 19742 arrives at some conclusion about the TOE's resistance to attacks. Table B.1 demonstrates the
 19743 relationship between this analysis and attack potential.

Vulnerability Component	TOE resistant to attacker with attack potential of:	Residual vulnerabilities only exploitable by attacker with attack potential of:
VAN.5	High	Beyond High
VAN.4	Moderate	High
VAN.3	Enhanced-Basic	Moderate
VAN.2	Basic	Enhanced-Basic
VAN.1	Basic	Enhanced-Basic

Table B.1 Vulnerability testing and attack potential

19744

19745 The “beyond high” entry in the residual vulnerabilities column of the above table represents those
 19746 potential vulnerabilities that would require an attacker to have an attack potential greater than
 19747 that of “high” in order to exploit the potential vulnerability. A vulnerability classified as residual in
 19748 this instance reflects the fact that a known weakness exists in the TOE, but in the current
 19749 operational environment, with the assumed attack potential, the weakness cannot be exploited.

19750 At any level of attack potential a potential vulnerability may be deemed “infeasible” due to a
 19751 countermeasure in the operational environment that prevents the vulnerability from being
 19752 exploited.

19753 A vulnerability analysis applies to all TSFI, including ones that access probabilistic or
 19754 permutational mechanisms. No assumptions are made regarding the correctness of the design and
 19755 implementation of the TSFI; nor are constraints placed on the attack method or the attacker's
 19756 interaction with the TOE - if an attack is possible, then it is to be considered during the
 19757 vulnerability analysis. As shown in Table B.1, successful evaluation against a vulnerability
 19758 assurance component reflects that the TSF is designed and implemented to protect against the
 19759 required level of threat.

19760 It is not necessary for an evaluator to perform an attack potential calculation for each potential
 19761 vulnerability. In some cases, it is apparent when developing the attack method whether or not the
 19762 attack potential required to develop and run the attack method is commensurate with that
 19763 assumed of the attacker in the operational environment. For any vulnerabilities for which an
 19764 exploitation is determined, the evaluator performs an attack potential calculation to determine that
 19765 the exploitation is appropriate to the level of attack potential assumed for the attacker.

19766 The approach described below is to be applied whenever it is necessary to calculate attack
19767 potential, unless the evaluation authority provides mandatory guidance that an alternative
19768 approach is to be applied. The values given in Tables B.2 and B.3 below are not mathematically
19769 proven. Therefore, the values given in these example tables may need to be adjusted according to
19770 the technology type and specific environments. The evaluator should seek guidance from the
19771 evaluation authority.

19772 **B.4 Calculating attack potential**

19773 **B.4.1 Application of attack potential**

19774 Attack potential is a function of expertise, resources and motivation. There are multiple methods of
19775 representing and quantifying these factors. Also, there may be other factors that are applicable for
19776 particular TOE types.

19777 **B.4.1.1 Treatment of motivation**

19778 Motivation is an attack potential factor that can be used to describe several aspects related to the
19779 attacker and the assets the attacker desires. Firstly, motivation can imply the likelihood of an attack
19780 - one can infer from a threat described as highly motivated that an attack is imminent, or that no
19781 attack is anticipated from an un-motivated threat. However, except for the two extreme levels of
19782 motivation, it is difficult to derive a probability of an attack occurring from motivation.

19783 Secondly, motivation can imply the value of the asset, monetarily or otherwise, to either the
19784 attacker or the asset holder. An asset of very high value is more likely to motivate an attack
19785 compared to an asset of little value. However, other than in a very general way, it is difficult to
19786 relate asset value to motivation because the value of an asset is subjective - it depends largely upon
19787 the value an asset holder places on it.

19788 Thirdly, motivation can imply the expertise and resources with which an attacker is willing to
19789 effect an attack. One can infer that a highly-motivated attacker is likely to acquire sufficient
19790 expertise and resources to defeat the measures protecting an asset. Conversely, one can infer that
19791 an attacker with significant expertise and resources is not willing to effect an attack using them if
19792 the attacker's motivation is low.

19793 During the course of preparing for and conducting an evaluation, all three aspects of motivation are
19794 at some point considered. The first aspect, likelihood of attack, is what may inspire a developer to
19795 pursue an evaluation. If the developer believes that the attackers are sufficiently motivated to
19796 mount an attack, then an evaluation can provide assurance of the ability of the TOE to thwart the
19797 attacker's efforts. Where the operational environment is well defined, for example in a system
19798 evaluation, the level of motivation for an attack may be known, and will influence the selection of
19799 countermeasures.

19800 Considering the second aspect, an asset holder may believe that the value of the assets (however
19801 measured) is sufficient to motivate attack against them. Once an evaluation is deemed necessary,
19802 the attacker's motivation is considered to determine the methods of attack that may be attempted,
19803 as well as the expertise and resources used in those attacks. Once examined, the developer is able
19804 to choose the appropriate assurance level, in particular the AVA requirement components,
19805 commensurate with the attack potential for the threats. During the course of the evaluation, and in
19806 particular as a result of completing the vulnerability assessment activity, the evaluator determines
19807 whether or not the TOE, operating in its operational environment, is sufficient to thwart attackers
19808 with the identified expertise and resources.

19809 It may be possible for a PP author to quantify the motivation of an attacker, as the PP author has
19810 greater knowledge of the operational environment in which the TOE (conforming to the
19811 requirements of the PP) is to be placed. Therefore, the motivation could form an explicit part of the
19812 expression of the attack potential in the PP, along with the necessary methods and measures to
19813 quantify the motivation.

19814 **B.4.2 Characterising attack potential**

19815 This subclause examines the factors that determine attack potential, and provides some guidelines
19816 to help remove some of the subjectivity from this aspect of the evaluation process.

19817 **B.4.2.1 Determining the attack potential**

19818 The determination of the attack potential for an attack corresponds to the identification of the
19819 effort required to create the attack, and to demonstrate that it can be successfully applied to the
19820 TOE (including setting up or building any necessary test equipment), thereby exploiting the
19821 vulnerability in the TOE. The demonstration that the attack can be successfully applied needs to
19822 consider any difficulties in expanding a result shown in the laboratory to create a useful attack. For
19823 example, where an experiment reveals some bits or bytes of a confidential data item (such as a key),
19824 it is necessary to consider how the remainder of the data item would be obtained (in this example
19825 some bits might be measured directly by further experiments, while others might be found by a
19826 different technique such as exhaustive search). It may not be necessary to carry out all of the
19827 experiments to identify the full attack, provided it is clear that the attack actually proves that
19828 access has been gained to a TOE asset, and that the complete attack could realistically be carried
19829 out in exploitation according to the AVA_VAN component targeted. In some cases, the only way to
19830 prove that an attack can realistically be carried out in exploitation according to the AVA_VAN
19831 component targeted is to perform completely the attack and then rate it based upon the resources
19832 actually required. One of the outputs from the identification of a potential vulnerability is assumed
19833 to be a script that gives a step-by-step description of how to carry out the attack that can be used in
19834 the exploitation of the vulnerability on another instance of the TOE.

19835 In many cases, the evaluators will estimate the parameters for exploitation, rather than carry out
19836 the full exploitation. The estimates and their rationale will be documented in the ETR.

19837 **B.4.2.2 Factors to be considered**

19838 The following factors should be considered during analysis of the attack potential required to
19839 exploit a vulnerability:

- 19840 a) Time taken to identify and exploit (*Elapsed Time*);
- 19841 b) Specialist technical expertise required (*Specialist Expertise*);
- 19842 c) Knowledge of the TOE design and operation (*Knowledge of the TOE*);
- 19843 d) Window of opportunity;
- 19844 e) IT hardware/software or other equipment required for exploitation.

19845 In many cases these factors are not independent, but may be substituted for each other in varying
19846 degrees. For example, expertise or hardware/software may be a substitute for time. A discussion of
19847 these factors follows. (The levels of each factor are discussed in increasing order of magnitude.)
19848 When it is the case, the less “expensive” combination is considered in the exploitation phase.

19849 *Elapsed time* is the total amount of time taken by an attacker to identify that a particular potential
19850 vulnerability may exist in the TOE, to develop an attack method and to sustain effort required to
19851 mount the attack against the TOE. When considering this factor, the worst-case scenario is used to
19852 estimate the amount of time required. The identified amount of time is as follows:

- 19853 a) less than one day;
- 19854 b) between one day and one week;
- 19855 c) between one week and two weeks;

- 19856 d) between two weeks and one month;
- 19857 e) each additional month up to 6 months leads to an increased value;
- 19858 f) more than 6 months.
- 19859 **Specialist expertise** refers to the level of generic knowledge of the underlying principles, product
 19860 type or attack methods (e.g. Internet protocols, Unix operating systems, buffer overflows). The
 19861 identified levels are as follows:
- 19862 a) Laymen are unknowledgeable compared to experts or proficient persons, with no
 19863 particular expertise;
- 19864 b) Proficient persons are knowledgeable in that they are familiar with the security
 19865 behaviour of the product or system type;
- 19866 c) Experts are familiar with the underlying algorithms, protocols, hardware, structures,
 19867 security behaviour, principles and concepts of security employed, techniques and tools
 19868 for the definition of new attacks, cryptography, classical attacks for the product type,
 19869 attack methods, etc. implemented in the product or system type.
- 19870 d) The level “Multiple Expert” is introduced to allow for a situation, where different fields of
 19871 expertise are required at an Expert level for distinct steps of an attack.
- 19872 It may occur that several types of expertise are required. By default, the higher of the different
 19873 expertises factors is chosen. In very specific cases, the “multiple expert” level could be used but it
 19874 should be noted that the expertise must concern fields that are strictly different like for example
 19875 HW manipulation and cryptography.
- 19876 **Knowledge of the TOE** refers to specific expertise in relation to the TOE. This is distinct from
 19877 generic expertise, but not unrelated to it. Identified levels are as follows:
- 19878 a) Public information concerning the TOE (e.g. as gained from the Internet);
- 19879 b) Restricted information concerning the TOE (e.g. knowledge that is controlled within the
 19880 developer organisation and shared with other organisations under a non-disclosure
 19881 agreement)
- 19882 c) Sensitive information about the TOE (e.g. knowledge that is shared between discreet
 19883 teams within the developer organisation, access to which is constrained only to members
 19884 of the specified teams);
- 19885 d) Critical information about the TOE (e.g. knowledge that is known by only a few
 19886 individuals, access to which is very tightly controlled on a strict need to know basis and
 19887 individual undertaking).
- 19888 The knowledge of the TOE may graduate according to design abstraction, although this can only be
 19889 done on a TOE by TOE basis. Some TOE designs may be public source (or heavily based on public
 19890 source) and therefore even the design representation would be classified as public or at most
 19891 restricted, while the implementation representation for other TOEs is very closely controlled as it
 19892 would give an attacker information that would aid an attack and is therefore considered to be
 19893 sensitive or even critical.
- 19894 It may occur that several types of knowledge are required. In such cases, the higher of the different
 19895 knowledge factors is chosen.
- 19896 **Window of opportunity** (Opportunity) is also an important consideration, and has a relationship
 19897 to the **Elapsed Time** factor. Identification or exploitation of a vulnerability may require

- 19898 considerable amounts of access to a TOE that may increase the likelihood of detection. Some attack
 19899 methods may require considerable effort off-line, and only brief access to the TOE to exploit. Access
 19900 may also need to be continuous, or over a number of sessions.
- 19901 For some TOEs the **Window of opportunity** may equate to the number of samples of the TOE that
 19902 the attacker can obtain. This is particularly relevant where attempts to penetrate the TOE and
 19903 undermine the SFRs may result in the destruction of the TOE preventing use of that TOE sample for
 19904 further testing, e.g. hardware devices. Often in these cases distribution of the TOE is controlled and
 19905 so the attacker must apply effort to obtain further samples of the TOE.
- 19906 For the purposes of this discussion:
- 19907 a) unnecessary/unlimited access means that the attack doesn't need any kind of opportunity
 19908 to be realised because there is no risk of being detected during access to the TOE and it is
 19909 no problem to access the number of TOE samples for the attack;
- 19910 b) easy means that access is required for less than a day and that the number of TOE
 19911 samples required to perform the attack is less than ten;
- 19912 c) moderate means that access is required for less than a month and that the number of TOE
 19913 samples required to perform the attack is less than one hundred;
- 19914 d) difficult means that access is required for at least a month or that the number of TOE
 19915 samples required to perform the attack is at least one hundred;
- 19916 e) none means that the opportunity window is not sufficient to perform the attack (the
 19917 length for which the asset to be exploited is available or is sensitive is less than the
 19918 opportunity length needed to perform the attack - for example, if the asset key is changed
 19919 each week and the attack needs two weeks); another case is, that a sufficient number of
 19920 TOE samples needed to perform the attack is not accessible to the attacker - for example
 19921 if the TOE is a hardware and the probability to destroy the TOE during the attack instead
 19922 of being successful is very high and the attacker has only access to one sample of the TOE.
- 19923 Consideration of this factor may result in determining that it is not possible to complete the exploit,
 19924 due to requirements for time availability that are greater than the opportunity time.
- 19925 **IT hardware/software or other equipment** refers to the equipment required to identify or exploit
 19926 a vulnerability.
- 19927 a) Standard equipment is readily available to the attacker, either for the identification of a
 19928 vulnerability or for an attack. This equipment may be a part of the TOE itself (e.g. a
 19929 debugger in an operating system), or can be readily obtained (e.g. Internet downloads,
 19930 protocol analyser or simple attack scripts).
- 19931 b) Specialised equipment is not readily available to the attacker, but could be acquired
 19932 without undue effort. This could include purchase of moderate amounts of equipment
 19933 (e.g. power analysis tools, use of hundreds of PCs linked across the Internet would fall
 19934 into this category), or development of more extensive attack scripts or programs. If
 19935 clearly different test benches consisting of specialised equipment are required for
 19936 distinct steps of an attack this would be rated as bespoke.
- 19937 c) Bespoke equipment is not readily available to the public as it may need to be specially
 19938 produced (e.g. very sophisticated software), or because the equipment is so specialised
 19939 that its distribution is controlled, possibly even restricted. Alternatively, the equipment
 19940 may be very expensive.
- 19941 d) The level "Multiple Bespoke" is introduced to allow for a situation, where different types
 19942 of bespoke equipment are required for distinct steps of an attack.

Specialist expertise and **Knowledge of the TOE** are concerned with the information required for persons to be able to attack a TOE. There is an implicit relationship between an attacker's expertise (where the attacker may be one or more persons with complementary areas of knowledge) and the ability to effectively make use of equipment in an attack. The weaker the attacker's expertise, the lower the potential to use equipment (IT hardware/software or other equipment). Likewise, the greater the expertise, the greater the potential for equipment to be used in the attack. Although implicit, this relationship between expertise and the use of equipment does not always apply, for instance, when environmental measures prevent an expert attacker's use of equipment, or when, through the efforts of others, attack tools requiring little expertise to be effectively used are created and freely distributed (e.g. via the Internet).

B.4.2.3 Calculation of attack potential

Table B.2 identifies the factors discussed in the previous subclause and associates numeric values with the total value of each factor.

Where a factor falls close to the boundary of a range the evaluator should consider use of an intermediate value to those in the table. For example, if twenty samples are required to perform the attack then a value between one and four may be selected for that factor, or if the design is based on a publicly available design but the developer has made some alterations then a value between zero and three should be selected according to the evaluator's view of the impact of those design changes. The table is intended as a guide.

The "***" specification in the table in considering **Window of Opportunity** is not to be seen as a natural progression from the timescales specified in the preceding ranges associated with this factor. This specification identifies that for a particular reason the potential vulnerability cannot be exploited in the TOE in its intended operational environment. For example, access to the TOE may be detected after a certain amount of time in a TOE with a known environment (i.e. in the case of a system) where regular patrols are completed, and the attacker could not gain access to the TOE for the required two weeks undetected. However, this would not be applicable to a TOE connected to the network where remote access is possible, or where the physical environment of the TOE is unknown.

Factor	Value
Elapsed Time	
<= one day	0
<= one week	1
<= two weeks	2
<= one month	4
<= two months	7
<= three months	10
<= four months	13
<= five months	15
<= six months	17
> six months	19
Expertise	
Layman	0
Proficient	3*(1)
Expert	6
Multiple experts	8
Knowledge of TOE	
Public	0
Restricted	3

Factor	Value
Sensitive	7
Critical	11
Window of Opportunity	
Unnecessary / unlimited access	0
Easy	1
Moderate	4
Difficult	10
None	**(2)
Equipment	
Standard	0
Specialised	4 ⁽³⁾
Bespoke	7
Multiple bespoke	9

- 19971 ⁽¹⁾ When several proficient persons are required to complete the attack path, the resulting level of
19972 expertise still remains “proficient” (which leads to a 3 rating).
19973 ⁽²⁾ Indicates that the attack path is not exploitable due to other measures in the intended operational
19974 environment of the TOE.
19975 ⁽³⁾ If clearly different test benches consisting of specialised equipment are required for distinct steps of an
19976 attack, this should be rated as bespoke.

19977 **Table B.2 Calculation of attack potential**

19978 To determine the resistance of the TOE to the potential vulnerabilities identified the following
19979 steps should be applied:

- 19980 a) Define the possible attack scenarios {AS1, AS2, ..., ASn} for the TOE in the operational
19981 environment.
- 19982 b) For each attack scenario, perform a theoretical analysis and calculate the relevant attack
19983 potential using Table B.2.
- 19984 c) For each attack scenario, if necessary, perform penetration tests in order to confirm or to
19985 disprove the theoretical analysis.
- 19986 d) Divide all attack scenarios {AS1, AS2, ..., ASn} into two groups:
- 19987 1) the attack scenarios having been successful (i.e. those that have been used to successfully
19988 undermine the SFRs), and
- 19989 2) the attack scenarios that have been demonstrated to be unsuccessful.
- 19990 e) For each successful attack scenario, apply Table B.3 and determine, whether there is a
19991 contradiction between the resistance of the TOE and the chosen AVA_VAN assurance
19992 component, see the last column of Table B.3.
- 19993 f) Should one contradiction be found, the vulnerability assessment will fail, e.g. the author of
19994 the ST chose the component AVA_VAN.5 and an attack scenario with an attack potential
19995 of 21 points (high) has broken the security of the TOE. In this case, the TOE is resistant to
19996 attacker with attack potential 'Moderate', this contradicts to AVA_VAN.5, hence, the
19997 vulnerability assessment fails.

19998 The "Values" column of Table B.3 indicates the range of attack potential values (calculated using
19999 Table B.2) of an attack scenario that results in the SFRs being undermined.

Values	Attack potential required to exploit scenario:	Meets assurance components:	Failure of components:
0-9	Basic	-	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
10-13	Enhanced-Basic	AVA_VAN.1, AVA_VAN.2	AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
14-19	Moderate	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3	AVA_VAN.4, AVA_VAN.5
20-24	High	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4	AVA_VAN.5
=>25	Beyond High	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5	-

Table B.3 Rating of vulnerabilities and TOE resistance

20000
20001 An approach such as this cannot take account of every circumstance or factor, but should give a
20002 better indication of the level of resistance to attack required to achieve the standard ratings. Other
20003 factors, such as the reliance on unlikely chance occurrences are not included in the basic model, but
20004 can be used by an evaluator as justification for a rating other than those that the basic model might
20005 indicate.

20006 It should be noted that whereas a number of vulnerabilities rated individually may indicate high
20007 resistance to attack, collectively the combination of vulnerabilities may indicate that overall a
20008 lower rating is applicable. The presence of one vulnerability may make another easier to exploit.

20009 If a PP/ST author wants to use the attack potential table for the determination of the level of attack
20010 the TOE should withstand (selection of Vulnerability analysis (AVA_VAN) component), they should
20011 proceed as follows: For all different attack scenarios (i.e. for all different types of attacker and/or
20012 different types of attack the author has in mind) which must not violate the SFRs, several passes
20013 through Table B.2 should be made to determine the different values of attack potential assumed for
20014 each such unsuccessful attack scenario. The PP/ST author then chooses the highest value of them
20015 in order to determine the level of the TOE resistance to be claimed from Table B.3: the TOE
20016 resistance must be at least equal to this highest value determined. For example, the highest value of
20017 attack potentials of all attack scenarios, which must not undermine the TOE security policy,
20018 determined in such a way is Moderate; hence, the TOE resistance shall be at least Moderate (i.e.
20019 Moderate or High); therefore, the PP/ST author can choose either AVA_VAN.4 (for Moderate) or
20020 AVA_VAN.5 (for High) as the appropriate assurance component.

20021 **B.5 Example calculation for direct attack**

20022 Mechanisms subject to direct attack are often vital for system security and developers often
20023 strengthen these mechanisms. As an example, a TOE might use a simple pass number

20024 authentication mechanism that can be overcome by an attacker who has the opportunity to
 20025 repeatedly guess another user's pass number. The system can strengthen this mechanism by
 20026 restricting pass numbers and their use in various ways. During the course of the evaluation an
 20027 analysis of this direct attack could proceed as follows:

20028 Information gleaned from the ST and design evidence reveals that identification and authentication
 20029 provides the basis upon which to control access to network resources from widely distributed
 20030 terminals. Physical access to the terminals is not controlled by any effective means. The duration of
 20031 access to a terminal is not controlled by any effective means. Authorised users of the system choose
 20032 their own pass numbers when initially authorised to use the system, and thereafter upon user
 20033 request. The system places the following restrictions on the pass numbers selected by the user:

20034 a) the pass number must be at least four and no greater than six digits long;

20035 b) consecutive numerical sequences are disallowed (such as 7,6,5,4,3);

20036 c) repeating digits is disallowed (each digit must be unique).

20037 Guidance provided to the users at the time of pass number selection is that pass numbers should be
 20038 as random as possible and should not be affiliated with the user in some way - a date of birth, for
 20039 instance.

20040 The pass number space is calculated as follows:

20041 a) Patterns of human usage are important considerations that can influence the approach to
 20042 searching a password space. Assuming the worst-case scenario and the user chooses a
 20043 number comprising only four digits, the number of pass number permutations assuming
 20044 that each digit must be unique is:

$$20045 \quad 7(8)(9)(10) = 5040$$

20046 b) The number of possible increasing sequences is seven, as is the number of decreasing
 20047 sequences. The pass number space after disallowing sequences is:

$$20048 \quad 5040 - 14 = 5026$$

20049 Based on further information gleaned from the design evidence, the pass number mechanism is
 20050 designed with a terminal locking feature. Upon the sixth failed authentication attempt the terminal
 20051 is locked for one hour. The failed authentication count is reset after five minutes so that an attacker
 20052 can at best attempt five pass number entries every five minutes, or 60 pass number entries every
 20053 hour.

20054 On average, an attacker would have to enter 2513 pass numbers, over 2513 minutes, before
 20055 entering the correct pass number. The average successful attack would, as a result, occur in slightly
 20056 less than:

$$20057 \quad \frac{2513 \text{ min}}{60 \frac{\text{min}}{\text{hour}}} \approx 42 \text{ hours}$$

20058 Using the approach to calculate the attack potential, described in the previous subclause, identifies
 20059 that it is possible that a layman can defeat the mechanism within days (given easy access to the
 20060 TOE), with the use of standard equipment, and with no knowledge of the TOE, giving a value of 1.
 20061 Given the resulting sum, 1, the attack potential required to effect a successful attack is not rated, as
 20062 it falls below that considered to be Basic.

Annex C

Evaluation Techniques and Tools (informative)

C.1 Semiformal and formal methods

In ISO/IEC 15408-3, Annex A.5, supplementary material on formal methods is provided.

C.1.1 Description of styles

This section provides general guidance on specification styles. Specific and detailed information is in those work units under the specific evaluator action elements where examination of the style of specifications, TSP model and correspondence demonstrations has to be performed.

The ADV class mandates three types of specification styles: informal, semiformal and formal. These styles are briefly described in the application notes to the ADV class in ISO/IEC 15408-2. The functional specification and design specification will be written using one or more of these specification styles. The TSF representations (in the following referred to as specifications) may use one or more notations in semiformal and formal style. The level of formality of the correspondence representation depends on the style of the adjacent pair of provided TSF representation (see the ADV_TDS family for details).

The hierarchy of components within these families increase the formality of the styles

- to reduce the ambiguity of the TSF representation through the hierarchy of components within the families,

- to reduce the likelihood of refinement errors in the available TSF representations,

- to strengthen the evidence for correctness of the TSF representations and the methods for their examination.

The styles are shortly characterised by

- informal style- defined semantics

- semiformal style - defined semantics and syntax

- formal style - defined semantics, syntax and rules of inference.

Regarding the notions of semantics and syntax the degree of precision varies with the style of description.

Informal descriptions require the semantics to provide meaning to all terms with the help of natural language explanations.

Semiformal descriptions restrict the syntax formation of terms to well defined expressions having a precise meaning in the technical community.

Formal style descriptions restrict the semantics and syntax even further: The formation of syntactical terms follows a formal language description required to be decidable. Examples include well established implicit formation rules being as precise as the formation of terms and formulas in first order predicate calculus or formal meta language descriptions using Extended Backus Naur Form. Apart from informal descriptions the semantics of formal terms is restricted to well established mathematical models. Formal derivation of theorems is restricted to predefined

20103 inference rules, which are based on well known logical reasoning (classical logic, intuitionistic logic,
 20104 modal logic, temporal logic, etc.). Algorithmic model checking can serve as a substitute for theorem
 20105 proving whenever the reference to well established model checkers is clear and appropriate meta
 20106 theorems are given to guarantee the equivalence to an inference by proof rules.

20107 In the context of the level of formality informal, semiformal and formal styles are considered to be
 20108 hierarchical in nature. Thus, requirements for a informal or semiformal style of specification may
 20109 also be met with either a semiformal or formal specification style provided, that is supported by
 20110 informal, explanatory text where appropriate. The set of presentation elements, syntactic and
 20111 semantic rules is referred in the following as notation. A formal style of presentation uses a formal
 20112 notation and rules of inference which is referred to in the following as formal system.

20113 The content and presentation elements of ADV_FSP and ADV_TDS compoentns describe the style
 20114 in which the presentation of evidence shall be provided by the developer. The evaluator action
 20115 element ADV_x.y.1E requires the evaluator to confirm that the information provided meets all
 20116 requirements for presentation of evidence. If the content and presentation elements require an
 20117 informal style the evaluator may perform the work units for the evaluator action elements in
 20118 parallel with the other work units examining the content of evidence. If the content and
 20119 presentation elements require a semiformal or a formal style this implies the application of
 20120 semiformal or formal methods to examine the content. Therefore it is recommended to perform
 20121 the work units for the evaluator action elements concerning the correct use of the method and its
 20122 rigour before the analysis of the content of evidence. If a notation or their usage in the
 20123 documentation does not provide the level of formality the necessary rigorous methods of analysis
 20124 may be not applicable. The work unit for the evaluator action elements examining the necessary
 20125 informal explanatory text may be performed in parallel with the other work units. Of course the
 20126 evaluator might detect errors in the presentation of evidence during the evaluator action as well
 20127 which result in a fail verdict for the evaluator action elements.

20128 The following text provides a guidance for the examination of specification styles and their use for
 20129 correspondence demonstration in the sub-activities for the assurance families ADV_FSP, ADV_TDS
 20130 and ADV_SPM.

20131

20132 Informal style

20133 An informal specification is one that is expressed in a natural language. If content and presentation
 20134 elements require an informal specification the work unit
 20135 for the evaluator action elements will require the evaluator to determine that it contains all
 20136 necessary informal explanatory text. The evaluator should examine the specification to make sure
 20137 that it

20138 - provides defined meanings of terms, abbreviations and acronyms that are
 20139 used in a context other than that accepted by normal usage,

20140 - if semiformal or formal notations are used appropriate informal, explanatory
 20141 text shall support the understanding.

20142 This enforces the informal specification to provide defined **semantics** of its statements. An
 20143 informal specification uses the ordinary conventions for the natural language i.e. any common
 20144 spoken tongue. It may use figures and semiformal elements of presentation like data flow diagrams
 20145 to illustrate the informal specification. If the specification uses a semiformal notation it will be
 20146 accompanied by supporting explanatory informal text appropriate for unambiguous common
 20147 understanding.

20148 Examples for the use of informal style are:

20149 - ISO/IEC 15408-1 identifies a glossary of terms specific to ISO/IEC
 20150 15408 and reserved terms in accordance with the ISO definitions contained in

- 20151 ISO/IEC Directives Part 2, Rules for the structure and drafting of International
20152 Standards. This clarifies the use of the verbs “shall”, “should”, “may” and “can” in
20153 the context of ISO/IEC 15408
- 20154 - International standards and the Request for Interpretation (RFC) are
20155 specified in an informal style. They use semiformal notations as well e.g. the
20156 abstract syntax notation ASN.1 for specification of message formats.
- 20157 Informal style does not excuse the absence of precision or informal definitions. The evaluator’s
20158 verdict fails if some technical term remains undefined, the evaluators lack of information prevents
20159 decision, or ambiguous interpretations cause confusion.
- 20160
- 20161 **Semiformal style**
- 20162 A semiformal specification is expressed in a restricted syntax language with defined semantics. It
20163 reduces the ambiguity of specification and strengthens the method of analysis.
- 20164 The evaluator should examine the identified notations to make sure that
- 20165 - The syntax rules are defined or a definition is referenced.
- 20166 - The notations with the explanatory text provide a defined **semantics** which
20167 is characterised by
- 20168 a) defined meanings of terms, abbreviations and acronyms that are used
20169 in a context other than that accepted by normal usage,
- 20170 b) the use of a semiformal notation is accompanied by supporting
20171 explanatory text in informal style appropriate for unambiguous meaning,
- 20172 c) expression of rules and characteristics of applicable policies, security
20173 functionality and interfaces (providing details of effects, exceptions and error
20174 messages) of TSF, their subsystems or modules to be specified for the assurance
20175 family for which the notations are used.
- 20176 - The notations contain a restricted **syntax** language which means
- 20177 d) a set of conventions must be supplied to define the restrictions
20178 imposed on the syntax.
- 20179 Examples for the use of semiformal style are:
- 20180 - The restricted syntax language may be a natural language with restricted
20181 sentence structure and keywords with special meanings. -> ISO/IEC 15408-1 and
20182 ISO/IEC 15408-2 provide a semiformal notation for the security functional
20183 requirements consisting of classes, families and components together with rules for
20184 permitted operations. As required by the ECD families of classes ASE and APE, an
20185 explicitly stated IT security requirement shall use the CC requirements components,
20186 families and classes as a model for presentation.
- 20187 - Formally specified languages may be used to define the data structures for
20188 the use of TSFI or an interface of subsystems or modules in semiformal style. Thus
20189 e.g. ISO/IEC 8824 and 8825 define the abstract syntax notation ASN.1 and ISO/IEC
20190 8834 the semantic of the object identifier (OID). ASN.1 makes possible extracting
20191 the encoded information by automated tools (parser). The interface specification
20192 may describe the complete details of all effects caused by interface usage by means
20193 of other semiformal notations e.g. state-transition diagrams.

- 20194 - Diagrams are commonly used for the specification of data-flow, state-
 20195 transition, entity-relation-ship, data or process or program structures in a semiformal
 20196 style, e.g. the Unified Modelling Language (UML) for object-oriented analysis and
 20197 design includes model diagrams, their semantics and an interchange format between
 20198 case tools. The graphical presentation assists the understanding of interaction and
 20199 behaviour of entities depending on events. The abstraction accompanied by the
 20200 graphical presentation normally needs to be compensated by informal description.
 20201 Data-flow and state-transition diagrams may be very helpful, e.g. for the precise
 20202 description and the analysis of protocols.
- 20203 - Programming languages like ANSI C defines a strong syntax and well-
 20204 defined semantics. The source code together with supporting explanatory text and
 20205 documentation of well-defined development tools provides an unambiguous
 20206 semiformal description of the TSF implementation, their security features and
 20207 interfaces. Although having a very high level of formality programming languages
 20208 may be of semiformal styles only because of missing inference rules. But some
 20209 software development tools support also formal methods in software design
 20210 including theorem prover.
- 20211 These examples show that semiformal style covers a wide range of capabilities and level of
 20212 formality. The developer should use appropriate notation for presentation of evidence depending
 20213 on the type of TOE (e.g. hardware, software), the development methodology and the purpose of the
 20214 specification.
- 20215 The semiformal style supports a structured analysis of the content, the consistency, the
 20216 completeness and the correspondence of the representation. A semiformal analysis is one that
 20217 results from a structured approach with a substantial degree of rigor in terms of completeness and
 20218 correctness.
- 20219 A semiformal interface specification supports the evaluator in analysing and assessing the external
 20220 behaviour of a TSF, their subsystems or modules for any input (e.g. to decide about acceptance or
 20221 rejection of a message and its content analysis). Semiformal evidence for conservation of
 20222 properties can be obtained by means of flow charts and state transition diagrams visualizing the
 20223 uniquely defined states and their interrelationship during the course of security preserving
 20224 transitions. The developer may use semiformal notations like software specification languages to
 20225 ensure correct refinement of the specifications from functional specification via high and low level
 20226 design down to the implementation level.
- 20227 This way the semiformal presentation clearly establishes its accuracy and superiority over
 20228 informal descriptions.

20229

20230 Formal style

- 20231 A formal specification is expressed within a formal system based upon well-established
 20232 mathematical concepts. These mathematical concepts are used to define well-defined semantics,
 20233 syntax and rules of inference. A formal system is an abstract system of identities and relations that
 20234 can be described by specifying a formal alphabet, a formal language over that alphabet which is
 20235 based on a formal syntax, and a set of formal rules of inference for constructing derivations of
 20236 sentences in the formal language.
- 20237 The evaluator should examine the identified formal systems to make sure that
- 20238 - The semantics, syntax and inference rules of the formal system are defined
 20239 or a definition is referenced.
- 20240 - Each formal system with the explanatory text provides a defined **semantics**
 20241 which

- 20242 a) provides defined meanings of terms, abbreviations and acronyms that
20243 are used in a context other than that accepted by normal usage,
- 20244 b) the use of a formal system and semiformal notation if any use is
20245 accompanied by supporting explanatory text in informal style appropriate for
20246 unambiguous meaning,
- 20247 c) the formal system is able to express rules and characteristics of
20248 applicable policies, security functionality and interfaces (providing details of
20249 effects, exceptions and error messages) of the TSF, their subsystems or modules to
20250 be specified for the assurance family for which the notations are used.
- 20251 d) the notation provides rules to determine the meaning of syntactical
20252 valid constructs.
- 20253 - Each formal system uses a formal **syntax** that
- 20254 e) provides rules to unambiguously recognise constructs.
- 20255 - Each formal system provides **proof rules** which
- 20256 f) support logical reasoning of well-established mathematical concepts,
- 20257 g) help to prevent derivation of contradictions.
- 20258 If the developer uses a formal system which is already accepted by the certification body the
20259 evaluator can rely on the level of formality and strength of the system and focus on the
20260 instantiation of the formal system to the TOE specifications and correspondence proofs.

20261 The formal style supports mathematical proofs of the security properties based on the security
20262 features, the consistency of refinements and the correspondence of the representations. Formal
20263 tool support seems adequate whenever manual derivations would otherwise become long winded
20264 and incomprehensible. Formal tools are also apt to reduce the error probability inherent in manual
20265 derivations.

20266 C.1.2 Security policy models and styles

20267 The assurance family Security policy modelling ADV_SPM requires in their components an
20268 increasing level of formality of the TSP model and correspondence demonstration between the TSP
20269 model and the functional specification. The following section provides some guidance how the
20270 general requirements on styles applies to the TSP models.

20271 The TOE Security Policy (TSP) is a set of rules and characteristics that regulate how assets are
20272 managed, protected and distributed within a TOE. The TSP can be explicitly stated in the ST by the
20273 SFR (e.g. of families FDP_ACC or FDP_AFC) or be drawn from other SFR (e.g. of classes FAU, FIA or
20274 FPR) claimed in the ST. Although these TSF are provided in semiformal style the policies are
20275 normally described by rules and characteristics in informal style. A TOE security policy model is a
20276 structured representation of security policies to be enforced by the TOE.

20277 According to ADV_SPM.*.2C the TSP model shall model all security policies of the TSP that can be
20278 modelled by the respective level defined by ADV_SPM.*.1C or a rationale shall be given why a lower
20279 level of formality is applied. Thus the TSP model may contain for policy sets of the TSP different
20280 models of different levels of formality as state of the art.

20281 An informal TSP model is a description of the TSP enforced by the security functional requirements
20282 claimed in the ST. All TSP in the ST can be informally modelled.

20283 Modelling means to describe the rules and characteristics of the policies by the properties and
20284 features in the TSP model and to provide evidence that the features imply these properties. The
20285 strength of this evidence depends on the level of formality: an informal model may provide a

20286 rationale but a formal model shall provide a formal proof that the security features imply the
 20287 security properties.

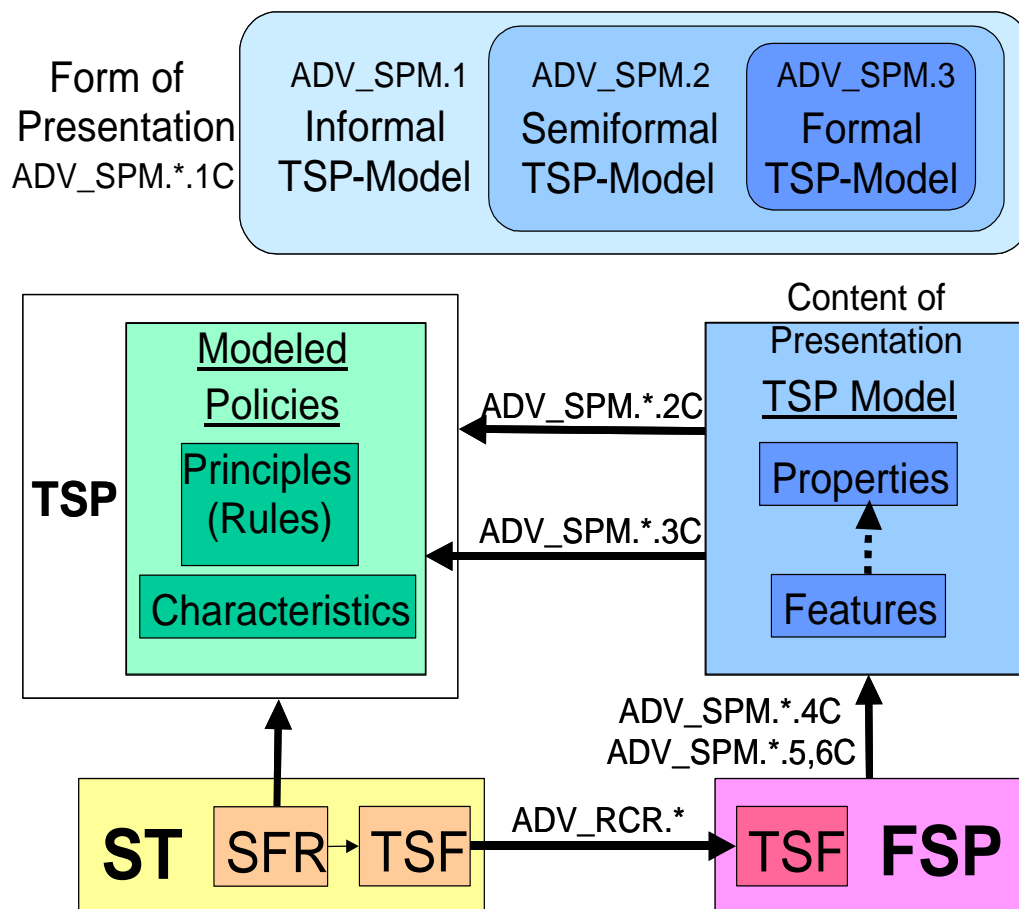


Figure 1.3 TOE security policy models and correspondence demonstration

20291 The possibility of formally modelling TSPs is dependent on the state of the art. A wide range of
 20292 examples have already been given in the past for successfully modelling Access Control including
 20293 Identification and Authentication. Hence inclusion of access control policies almost always requires
 20294 the developer to provide the model in a formal style.

20295 Whenever in doubt the evaluator should negotiate the type of style (formal, semiformal or
 20296 informal) with the certification body in advance in order to agree upon the state of the art for the
 20297 specific policy under question.

20298

20299

20300

20301

20302

20303

20304

20305
20306
20307
20308
20309
20310
20311
20312
20313
20314
20315
20316
20317
20318