Common Methodology for Information Technology Security Evaluation

Evaluation methodology

September 2006

Version 3.1
Revision 1

CCMB-2006-09-004

Foreword

This version of the Common Methodology for Information Technology Security Evaluation (CEM v3.1) is the first major revision since being published as CEM v2.3 in 2005.

CEM v3.1 aims to: eliminate redundant evaluation activities; reduce/eliminate activities that contribute little to the final assurance of a product; clarify CEM terminology to reduce misunderstanding; restructure and refocus the evaluation activities to those areas where security assurance is gained; and add new CEM requirements if needed.

Trademarks:

– UNIX is a registered trademark of The Open Group in the United States and other countries

– Windows is a registered trademark of Microsoft Corporation in the United States and other countries

Legal Notice:

The governmental organisations listed below contributed to the development of this version of the Common Methodology for Information Technology Security Evaluation. As the joint holders of the copyright in the Common Methodology for Information Technology Security Evaluation, version 3.1 *(called "CEM* 3.1*"), they hereby grant non-exclusive license to ISO/IEC to use CEM* 3.1 *in the continued development/maintenance of the ISO/IEC 18045 international standard. However, these governmental organisations retain the right to use, copy, distribute, translate or modify CEM* 3.1 *as they see fit.*

Australia/New Zealand:	*The Defence Signals Directorate and the Government Communications Security Bureau respectively;*
Canada:	*Communications Security Establishment;*
France:	*Direction Centrale de la Sécurité des Systèmes d'Information;*
Germany:	*Bundesamt für Sicherheit in der Informationstechnik;*
Japan:	*Information Technology Promotion Agency*
Netherlands:	*Netherlands National Communications Security Agency;*
Spain:	*Ministerio de Administraciones Públicas and Centro Criptológico Nacional;*
United Kingdom:	*Communications-Electronics Security Group;*
United States:	*The National Security Agency and the National Institute of Standards and Technology.*

Table of Contents

1 INTRODUCTION 12

2 SCOPE 13

3 NORMATIVE REFERENCES 14

4 TERMS AND DEFINITIONS 15

5 SYMBOLS AND ABBREVIATED TERMS 17

6 OVERVIEW 18

6.1 Organisation of the CEM 18

7 DOCUMENT CONVENTIONS 19

7.1 Terminology 19

7.2 Verb usage 19

7.3 General evaluation guidance 19

7.4 Relationship between CC and CEM structures 19

8 EVALUATION PROCESS AND RELATED TASKS 21

8.1 Introduction 21

8.2 Evaluation process overview 21
8.2.1 Objectives 21
8.2.2 Responsibilities of the roles 21
8.2.3 Relationship of roles 22
8.2.4 General evaluation model 22
8.2.5 Evaluator verdicts 23

8.3 Evaluation input task 25
8.3.1 Objectives 25
8.3.2 Application notes 25
8.3.3 Management of evaluation evidence sub-task 26

8.4 Evaluation sub-activities 27

8.5 Evaluation output task 27
8.5.1 Objectives 27
8.5.2 Management of evaluation outputs 27
8.5.3 Application notes 28
8.5.4 Write OR sub-task 28
8.5.5 Write ETR sub-task 28

9 CLASS APE: PROTECTION PROFILE EVALUATION 35

Table of contents

9.1 Introduction 35

9.2 Application notes 35
9.2.1 Re-using the evaluation results of certified PPs 35

9.3 PP introduction (APE_INT) 36
9.3.1 Evaluation of sub-activity (APE_INT.1) 36

9.4 Conformance claims (APE_CCL) 38
9.4.1 Evaluation of sub-activity (APE_CCL.1) 38

9.5 Security problem definition (APE_SPD) 43
9.5.1 Evaluation of sub-activity (APE_SPD.1) 43

9.6 Security objectives (APE_OBJ) 45
9.6.1 Evaluation of sub-activity (APE_OBJ.1) 45
9.6.2 Evaluation of sub-activity (APE_OBJ.2) 45

9.7 Extended components definition (APE_ECD) 49
9.7.1 Evaluation of sub-activity (APE_ECD.1) 49

9.8 Security requirements (APE_REQ) 54
9.8.1 Evaluation of sub-activity (APE_REQ.1) 54
9.8.2 Evaluation of sub-activity (APE_REQ.2) 57

10 CLASS ASE: SECURITY TARGET EVALUATION 63

10.1 Introduction 63

10.2 Application notes 63
10.2.1 Re-using the evaluation results of certified PPs 63

10.3 ST introduction (ASE_INT) 64
10.3.1 Evaluation of sub-activity (ASE_INT.1) 64

10.4 Conformance claims (ASE_CCL) 67
10.4.1 Evaluation of sub-activity (ASE_CCL.1) 67

10.5 Security problem definition (ASE_SPD) 75
10.5.1 Evaluation of sub-activity (ASE_SPD.1) 75

10.6 Security objectives (ASE_OBJ) 77
10.6.1 Evaluation of sub-activity (ASE_OBJ.1) 77
10.6.2 Evaluation of sub-activity (ASE_OBJ.2) 77

10.7 Extended components definition (ASE_ECD) 81
10.7.1 Evaluation of sub-activity (ASE_ECD.1) 81

10.8 Security requirements (ASE_REQ) 86
10.8.1 Evaluation of sub-activity (ASE_REQ.1) 86
10.8.2 Evaluation of sub-activity (ASE_REQ.2) 89

10.9 TOE summary specification (ASE_TSS) 95
10.9.1 Evaluation of sub-activity (ASE_TSS.1) 95
10.9.2 Evaluation of sub-activity (ASE_TSS.2) 95

11 CLASS ADV: DEVELOPMENT 98

11.1 Introduction 98

11.2 Application notes 98

11.3 Security Architecture (ADV_ARC) 100
11.3.1 Evaluation of sub-activity (ADV_ARC.1) 100

11.4 Functional specification (ADV_FSP) 106
11.4.1 Evaluation of sub-activity (ADV_FSP.1) 106
11.4.2 Evaluation of sub-activity (ADV_FSP.2) 110
11.4.3 Evaluation of sub-activity (ADV_FSP.3) 115
11.4.4 Evaluation of sub-activity (ADV_FSP.4) 122
11.4.5 Evaluation of sub-activity (ADV_FSP.5) 128
11.4.6 Evaluation of sub-activity (ADV_FSP.6) 135

11.5 Implementation representation (ADV_IMP) 136
11.5.1 Evaluation of sub-activity (ADV_IMP.1) 136
11.5.2 Evaluation of sub-activity (ADV_IMP.2) 138

11.6 TSF internals (ADV_INT) 139
11.6.1 Evaluation of sub-activity (ADV_INT.1) 139
11.6.2 Evaluation of sub-activity (ADV_INT.2) 142
11.6.3 Evaluation of sub-activity (ADV_INT.3) 144

11.7 Security policy modelling (ADV_SPM) 145
11.7.1 Evaluation of sub-activity (ADV_SPM.1) 145

11.8 TOE design (ADV_TDS) 146
11.8.1 Evaluation of sub-activity (ADV_TDS.1) 146
11.8.2 Evaluation of sub-activity (ADV_TDS.2) 150
11.8.3 Evaluation of sub-activity (ADV_TDS.3) 156
11.8.4 Evaluation of sub-activity (ADV_TDS.4) 168
11.8.5 Evaluation of sub-activity (ADV_TDS.5) 179
11.8.6 Evaluation of sub-activity (ADV_TDS.6) 179

12 CLASS AGD: GUIDANCE DOCUMENTS 180

12.1 Introduction 180

12.2 Application notes 180

12.3 Operational user guidance (AGD_OPE) 181
12.3.1 Evaluation of sub-activity (AGD_OPE.1) 181

12.4 Preparative procedures (AGD_PRE) 185
12.4.1 Evaluation of sub-activity (AGD_PRE.1) 185

13 CLASS ALC: LIFE-CYCLE SUPPORT 188

13.1 Introduction 188

13.2 CM capabilities (ALC_CMC) 189
13.2.1 Evaluation of sub-activity (ALC_CMC.1) 189
13.2.2 Evaluation of sub-activity (ALC_CMC.2) 190
13.2.3 Evaluation of sub-activity (ALC_CMC.3) 192
13.2.4 Evaluation of sub-activity (ALC_CMC.4) 196
13.2.5 Evaluation of sub-activity (ALC_CMC.5) 203

Table of contents

13.3 CM scope (ALC_CMS) 212
13.3.1 Evaluation of sub-activity (ALC_CMS.1) 212
13.3.2 Evaluation of sub-activity (ALC_CMS.2) 212
13.3.3 Evaluation of sub-activity (ALC_CMS.3) 213
13.3.4 Evaluation of sub-activity (ALC_CMS.4) 214
13.3.5 Evaluation of sub-activity (ALC_CMS.5) 215

13.4 Delivery (ALC_DEL) 218
13.4.1 Evaluation of sub-activity (ALC_DEL.1) 218

13.5 Development security (ALC_DVS) 220
13.5.1 Evaluation of sub-activity (ALC_DVS.1) 220
13.5.2 Evaluation of sub-activity (ALC_DVS.2) 223

13.6 Flaw remediation (ALC_FLR) 228
13.6.1 Evaluation of sub-activity (ALC_FLR.1) 228
13.6.2 Evaluation of sub-activity (ALC_FLR.2) 230
13.6.3 Evaluation of sub-activity (ALC_FLR.3) 234

13.7 Life-cycle definition (ALC_LCD) 241
13.7.1 Evaluation of sub-activity (ALC_LCD.1) 241
13.7.2 Evaluation of sub-activity (ALC_LCD.2) 242

13.8 Tools and techniques (ALC_TAT) 245
13.8.1 Evaluation of sub-activity (ALC_TAT.1) 245
13.8.2 Evaluation of sub-activity (ALC_TAT.2) 247
13.8.3 Evaluation of sub-activity (ALC_TAT.3) 250

14 CLASS ATE: TESTS 254

14.1 Introduction 254

14.2 Application notes 254
14.2.1 Understanding the expected behaviour of the TOE 255
14.2.2 Testing vs. alternate approaches to verify the expected behaviour of functionality 255
14.2.3 Verifying the adequacy of tests 256

14.3 Coverage (ATE_COV) 257
14.3.1 Evaluation of sub-activity (ATE_COV.1) 257
14.3.2 Evaluation of sub-activity (ATE_COV.2) 258
14.3.3 Evaluation of sub-activity (ATE_COV.3) 259

14.4 Depth (ATE_DPT) 260
14.4.1 Evaluation of sub-activity (ATE_DPT.1) 260
14.4.2 Evaluation of sub-activity (ATE_DPT.2) 262
14.4.3 Evaluation of sub-activity (ATE_DPT.3) 265
14.4.4 Evaluation of sub-activity (ATE_DPT.4) 267

14.5 Functional tests (ATE_FUN) 268
14.5.1 Evaluation of sub-activity (ATE_FUN.1) 268
14.5.2 Evaluation of sub-activity (ATE_FUN.2) 271

14.6 Independent testing (ATE_IND) 272
14.6.1 Evaluation of sub-activity (ATE_IND.1) 272
14.6.2 Evaluation of sub-activity (ATE_IND.2) 277
14.6.3 Evaluation of sub-activity (ATE_IND.3) 283

15 CLASS AVA: VULNERABILITY ASSESSMENT 284

15.1 Introduction 284

15.2 Vulnerability analysis (AVA_VAN) 285
15.2.1 Evaluation of sub-activity (AVA_VAN.1) 285
15.2.2 Evaluation of sub-activity (AVA_VAN.2) 291
15.2.3 Evaluation of sub-activity (AVA_VAN.3) 299
15.2.4 Evaluation of sub-activity (AVA_VAN.4) 310
15.2.5 Evaluation of sub-activity (AVA_VAN.5) 319

16 CLASS ACO: COMPOSITION 320

16.1 Introduction 320

16.2 Application notes 320

16.3 Composition rationale (ACO_COR) 322
16.3.1 Evaluation of sub-activity (ACO_COR.1) 322

16.4 Development evidence (ACO_DEV) 330
16.4.1 Evaluation of sub-activity (ACO_DEV.1) 330
16.4.2 Evaluation of sub-activity (ACO_DEV.2) 331
16.4.3 Evaluation of sub-activity (ACO_DEV.3) 333

16.5 Reliance of dependent component (ACO_REL) 337
16.5.1 Evaluation of sub-activity (ACO_REL.1) 337
16.5.2 Evaluation of sub-activity (ACO_REL.2) 339

16.6 Composed TOE testing (ACO_CTT) 343
16.6.1 Evaluation of sub-activity (ACO_CTT.1) 343
16.6.2 Evaluation of sub-activity (ACO_CTT.2) 346

16.7 Composition vulnerability analysis (ACO_VUL) 351
16.7.1 Evaluation of sub-activity (ACO_VUL.1) 351
16.7.2 Evaluation of sub-activity (ACO_VUL.2) 354
16.7.3 Evaluation of sub-activity (ACO_VUL.3) 359

A GENERAL EVALUATION GUIDANCE 364

A.1 Objectives 364

A.2 Sampling 364

A.3 Dependencies 366
A.3.1 Dependencies between activities 367
A.3.2 Dependencies between sub-activities 367
A.3.3 Dependencies between actions 367

A.4 Site Visits 368
A.4.1 Introduction 368
A.4.2 General Approach 368
A.4.3 Orientation Guide for the Preparation of the Check List 369
A.4.4 Example of a checklist 371

A.5 Scheme Responsibilities 374

B VULNERABILITY ASSESSMENT (AVA) 377

Table of contents

B.1 What is Vulnerability Analysis 377

B.2 Evaluator construction of a Vulnerability Analysis 378
B.2.1 Generic vulnerability guidance 378
B.2.2 Identification of Potential Vulnerabilities 388

B.3 When attack potential is used 392
B.3.1 Developer 392
B.3.2 Evaluator 392

B.4 Calculating attack potential 394
B.4.1 Application of attack potential 394
B.4.2 Characterising attack potential 395

B.5 Example calculation for direct attack 404

List of figures

Figure 1 - Mapping of the CC and CEM structures 20
Figure 2 - Generic evaluation model 23
Figure 3 - Example of the verdict assignment rule 24
Figure 4 - ETR information content for a PP evaluation 29
Figure 5 - ETR information content for a TOE evaluation 32

List of tables

Table 1 Example of a checklist at EAL 4 (extract) 373
Table 2 Vulnerability testing and attack potential 393
Table 3 Calculation of attack potential 401
Table 4 Rating of vulnerabilities and TOE resistance 403

1 Introduction

1 The target audience for the Common Methodology for Information Technology Security Evaluation (CEM) is primarily evaluators applying the CC and certifiers confirming evaluator actions; evaluation sponsors, developers, PP/ST authors and other parties interested in IT security may be a secondary audience.

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

2 Scope

3 The Common Methodology for Information Technology Security Evaluation (CEM) is a companion document to the Common Criteria for Information Technology Security Evaluation (CC). The CEM describes the minimum actions to be performed by an evaluator in order to conduct a CC evaluation, using the criteria and evaluation evidence defined in the CC.

3 Normative references

4 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.

CC Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 1, September 2006.

4 Terms and definitions

5 For the purposes of this document, the following terms and definitions apply.

6 Terms which are presented in bold-faced type are themselves defined in this Section.

7 **action —** evaluator action element of the CC Part 3. These actions are either explicitly stated as evaluator actions or implicitly derived from developer actions (implied evaluator actions) within the CC Part 3 assurance components.

8 **activity —** the application of an assurance class of the CC Part 3.

9 **check —** to generate a **verdict** by a simple comparison. Evaluator expertise is not required. The statement that uses this verb describes what is mapped.

10 **evaluation deliverable —** any resource required from the sponsor or developer by the evaluator or overseer to perform one or more evaluation or evaluation oversight activities.

11 **evaluation evidence —** a tangible **evaluation deliverable**.

12 **evaluation technical report —** a report that documents the **overall verdict** and its justification, produced by the evaluator and submitted to an overseer.

13 **examine —** to generate a **verdict** by analysis using evaluator expertise. The statement that uses this verb identifies what is analysed and the properties for which it is analysed.

14 **interpretation —** a clarification or amplification of a CC, CEM or **scheme** requirement.

15 **methodology —** the system of principles, procedures and processes applied to IT security evaluations.

16 **observation report —** a report written by the evaluator requesting a clarification or identifying a problem during the evaluation.

17 **overall verdict —** a *pass or fail* statement issued by an evaluator with respect to the result of an evaluation.

18 **oversight verdict —** a statement issued by an overseer confirming or rejecting an *overall verdict* based on the results of evaluation oversight activities.

19 **record —** to retain a written description of procedures, events, observations, insights and results in sufficient detail to enable the work performed during the evaluation to be reconstructed at a later time.

20 **report —** to include evaluation results and supporting material in the **Evaluation Technical Report** or an **Observation Report**.

21 **scheme —** set of rules, established by an evaluation authority, defining the evaluation environment, including criteria and **methodology** required to conduct IT security evaluations.

22 **sub-activity —** the application of an assurance component of the CC Part 3. Assurance families are not explicitly addressed in the CEM because evaluations are conducted on a single assurance component from an assurance family.

23 **tracing —** a simple directional relation between two sets of entities, which shows which entities in the first set correspond to which entities in the second.

24 **verdict —** a *pass, fail or inconclusive* statement issued by an evaluator with respect to a CC evaluator action element, assurance component, or class. Also see **overall verdict**.

25 **work unit —** the most granular level of evaluation work. Each CEM action comprises one or more work units, which are grouped within the CEM action by CC content and presentation of evidence or developer action element. The work units are presented in the CEM in the same order as the CC elements from which they are derived. Work units are identified in the left margin by a symbol such as ALC_TAT.1-2. In this symbol, the string *ALC_TAT.1* indicates the CC component (i.e. the CEM sub-activity), and the final digit (*2*) indicates that this is the second work unit in the ALC_TAT.1 sub-activity.

5 Symbols and abbreviated terms

CEM	Common Methodology for Information Technology Security Evaluation
ETR	Evaluation Technical Report
OR	Observation Report

6 Overview

6.1 Organisation of the CEM

26 Chapter 7 defines the conventions used in the CEM.

27 Chapter 8 describes general evaluation tasks with no verdicts associated with them as they do not map to CC evaluator action elements.

28 Chapter 9 addresses the work necessary for reaching an evaluation result on a PP.

29 Chapters 10 to 16 define the evaluation activities, organised by Assurance Classes.

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

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

7 Document Conventions

7.1 Terminology

32 Unlike the CC, where each element maintains the last digit of its identifying symbol for all components within the family, the CEM may introduce new work units when a CC evaluator action element changes from sub-activity to sub-activity; as a result, the last digit of the work unit's identifying symbol may change although the work unit remains unchanged.

33 Any methodology-specific evaluation work required that is not derived directly from CC requirements is termed *task* or *sub-task*.

7.2 Verb usage

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

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

36 The verbs *check*, *examine*, *report* and *record* are used with a precise meaning within this part of the CEM and the Chapter 4 should be referenced for their definitions.

7.3 General evaluation guidance

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

7.4 Relationship between CC and CEM structures

38 There are direct relationships between the CC structure (i.e. class, family, component and element) and the structure of the CEM. Figure 1 illustrates the correspondence between the CC constructs of class, family and evaluator action elements and CEM activities, sub-activities and actions. However,

several CEM work units may result from the requirements noted in CC developer action and content and presentation elements.

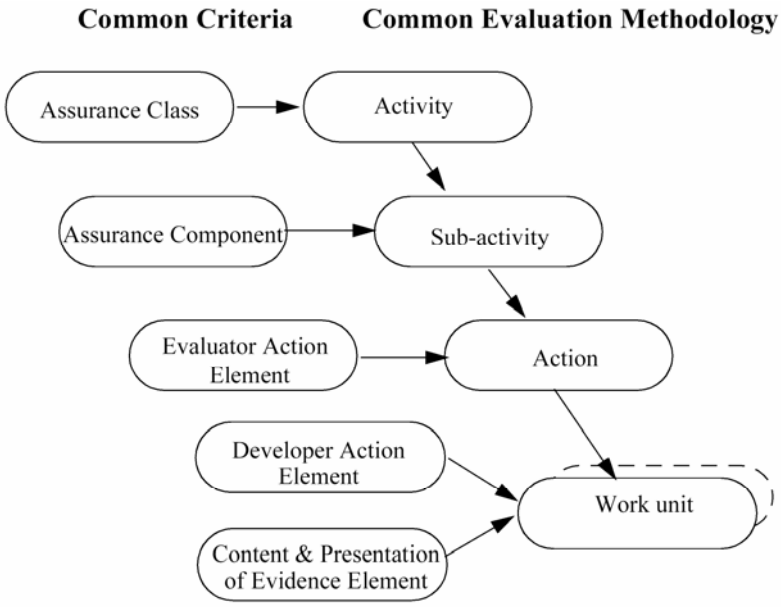

Figure 1 - Mapping of the CC and CEM structures

8 Evaluation process and related tasks

8.1 Introduction

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

40 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.

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

42 The evaluation sub-activities are only introduced in this chapter, and fully described in the following chapters.

43 In contrast to the evaluation sub-activities, input and output tasks have no verdicts associated with them as they do not map to CC evaluator action elements; they are performed in order to ensure conformance with the universal principles and to comply with the CEM.

44 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.

8.2 Evaluation process overview

8.2.1 Objectives

45 This section 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.

8.2.2 Responsibilities of the roles

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

47 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.

48 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.

49 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.

50 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.

8.2.3 Relationship of roles

51 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.

52 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.

8.2.4 General evaluation model

53 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.

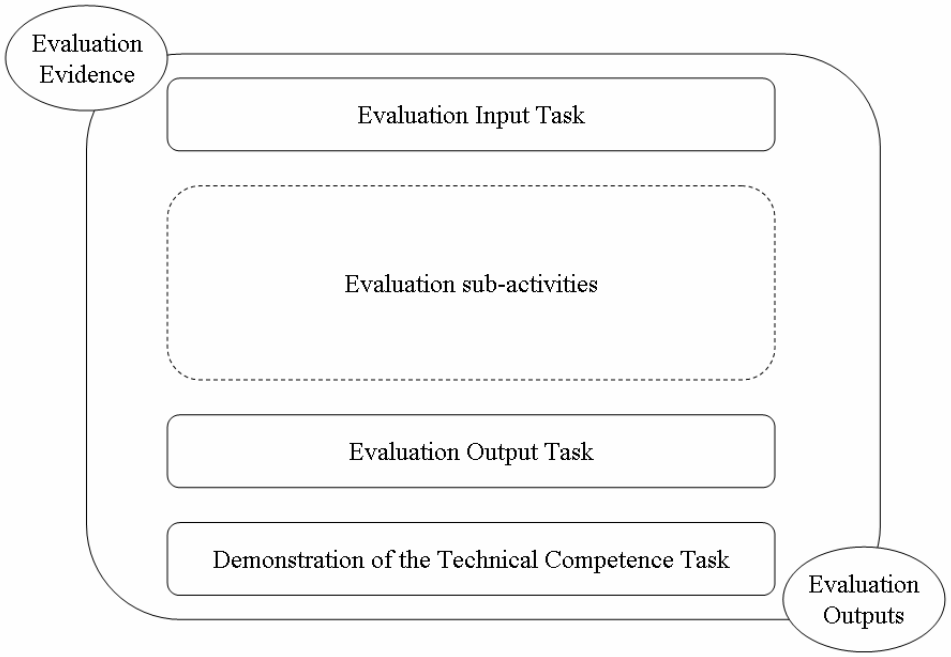

Figure 2 - Generic evaluation model

54 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.

8.2.5 Evaluator verdicts

55 The evaluator assigns verdicts to the requirements of the CC and not to those of the CEM. The most granular CC structure to which a verdict is assigned is the evaluator action element (explicit or implied). A verdict is assigned to an applicable CC evaluator action element as a result of performing the corresponding CEM action and its constituent work units. Finally, an evaluation result is assigned, as described in CC Part 1, Chapter 9, Evaluation results.

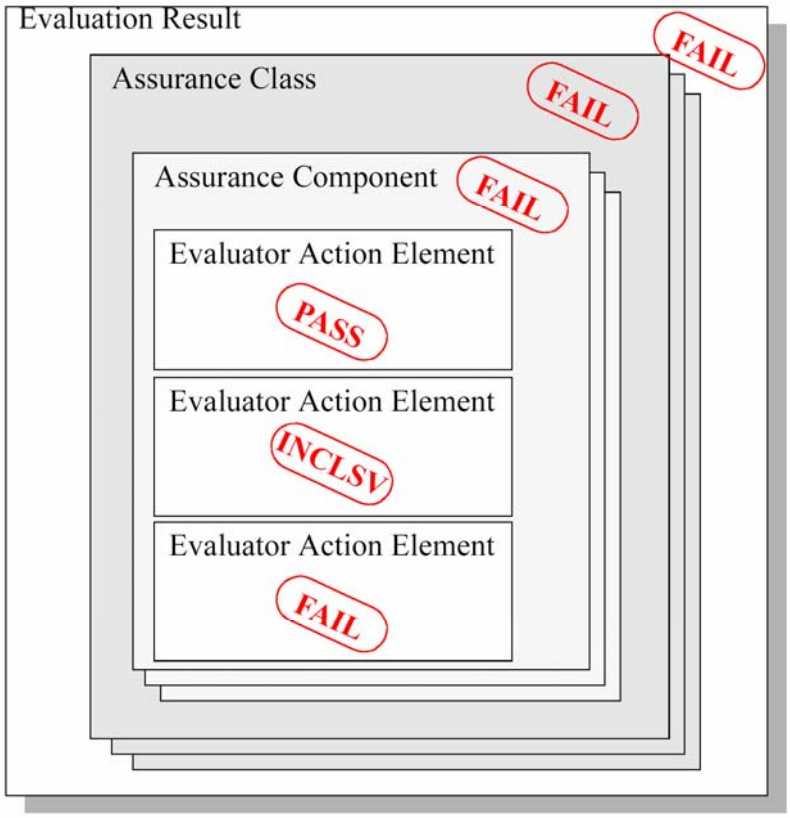

Figure 3 - Example of the verdict assignment rule

56 The CEM recognises three mutually exclusive verdict states:

a) Conditions for a *pass* verdict are defined as an evaluator completion of the CC 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 CEM 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 the CC 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.

57 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*.

8.3 Evaluation input task

8.3.1 Objectives

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

8.3.2 Application notes

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

60 Since the assurance requirements apply to the entire TOE, all evaluation evidence pertaining to all parts of the TOE is to be made available to the evaluator. The scope and required content of such evaluation evidence is independent of the level of control that the developer has over each of the parts of the TOE. For example, if design is required, then the TOE design (ADV_TDS) requirements will apply to all subsystems that are part of the TSF. In addition, assurance requirements that call for procedures to be in place (for example, CM capabilities (ALC_CMC) and Delivery (ALC_DEL)) will also apply to the entire TOE (including any part produced by another developer).

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

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

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

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

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

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

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

8.3.3 Management of evaluation evidence sub-task

8.3.3.1 Configuration control

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

66 The CC implies that the evaluator is able to identify and locate each item of evaluation evidence after it has been received and is able to determine whether a specific version of a document is in the evaluator's possession.

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

8.3.3.2 Disposal

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

a) returning the evaluation evidence;

b) archiving the evaluation evidence;

c) destroying the evaluation evidence.

8.3.3.3 Confidentiality

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

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

8.4 Evaluation sub-activities

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

8.5 Evaluation output task

8.5.1 Objectives

72 The objective of this Section is to describe the Observation Report (OR) and the Evaluation Technical Report (ETR). Schemes may require additional evaluator reports such as reports on individual units of work, or may require additional information to be contained in the OR and the ETR. The CEM does not preclude the addition of information into these reports as the CEM specifies only the minimum information content.

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

74 The evaluator performs the two following sub-tasks in order to achieve the CEM requirements for the information content of reports:

a) write OR sub-task (if needed in the context of the evaluation);

b) write ETR sub-task.

8.5.2 Management of evaluation outputs

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

8.5.3 Application notes

76 In this version of the CEM, 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.

8.5.4 Write OR sub-task

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

78 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.

79 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.

80 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 overseers and sponsors).

8.5.5 Write ETR sub-task

8.5.5.1 Objectives

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

82 The CEM 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 Chapters) be reported in the ETR.

83 The reader of the ETR is assumed to be familiar with general concepts of information security, the CC, the CEM, evaluation approaches and IT.

84 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 the CEM.

8.5.5.2 ETR for a PP Evaluation

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

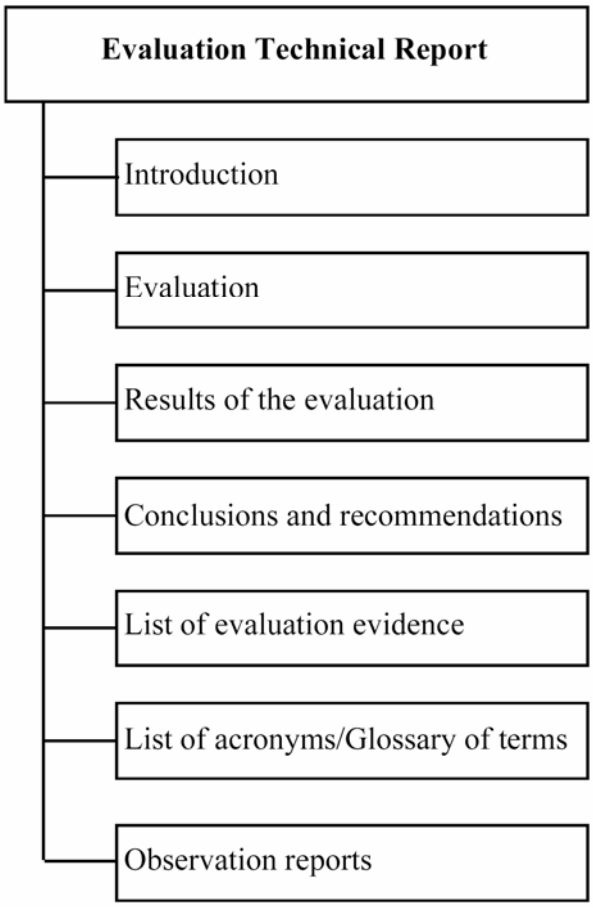

Figure 4 - ETR information content for a PP evaluation

8.5.5.2.1 Introduction

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

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

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

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

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

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

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

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

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

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

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

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

8.5.5.2.2 Evaluation

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

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

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

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

8.5.5.2.3 Results of the evaluation

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

103 The rationale justifies the verdict using the CC, the CEM, any interpretations and the evaluation evidence examined and shows how the evaluation evidence does or does not meet each aspect of the criteria. It contains a description of the work performed, the method used, and any derivation of results. The rationale may provide detail to the level of a CEM work unit.

8.5.5.2.4 Conclusions and recommendations

104 The evaluator ***shall report*** the conclusions of the evaluation, in particular the overall verdict as defined in CC Part 1 Chapter 9, Evaluation results, and determined by application of the verdict assignment described in 8.2.5.

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

8.5.5.2.5 List of evaluation evidence

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

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

– the title;

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

8.5.5.2.6 List of acronyms/Glossary of terms

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

108 Glossary definitions already defined by the CC or CEM need not be repeated in the ETR.

8.5.5.2.7 Observation reports

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

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

8.5.5.3 ETR for a TOE Evaluation

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

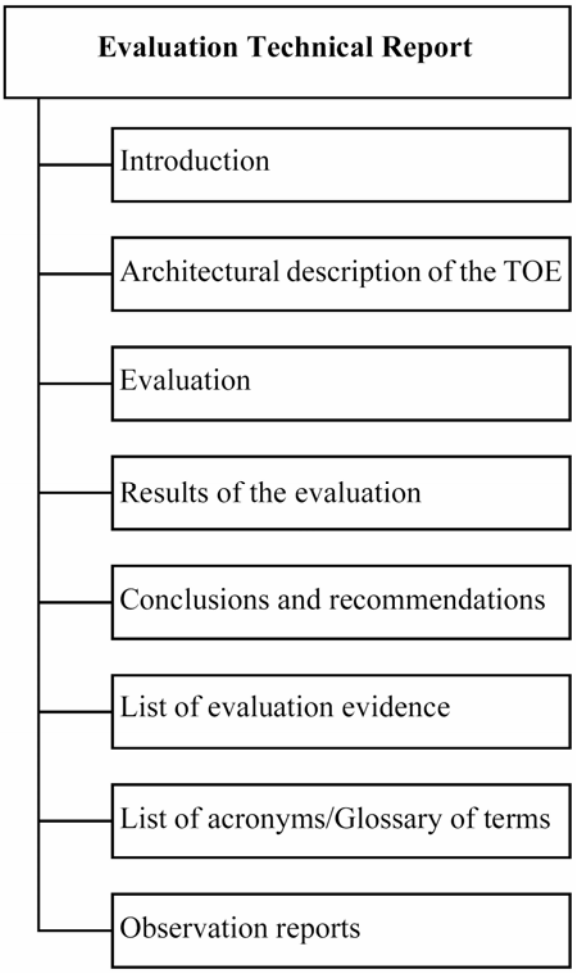

Figure 5 - ETR information content for a TOE evaluation

8.5.5.3.1 Introduction

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

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

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

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

116 The evaluator ***shall report*** ST and TOE configuration control identifiers.

117 ST and TOE configuration control identifiers identify what is being evaluated in order for the overseer to verify that the verdicts have been assigned correctly by the evaluator.

118 If the ST claims that the TOE conforms to the requirements of one or more PPs, the ETR shall report the reference of the corresponding PPs.

119 The PPs reference contains information that uniquely identifies the PPs (e.g. title, date, and version number).

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

121 The identity of the TOE developer is required to identify the party responsible for producing the TOE.

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

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

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

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

8.5.5.3.2 Architectural description of the TOE

126 The evaluator ***shall report*** a high level description of the TOE and its major components based on the evaluation evidence described in the CC assurance family entitled TOE design (ADV_TDS), where applicable.

127 The intent of this Section is to characterise the degree of architectural separation of the major components. If there is no TOE design (ADV_TDS) requirement in the ST, this is not applicable and is considered to be satisfied.

8.5.5.3.3 Evaluation

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

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

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

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

8.5.5.3.4 Results of the evaluation

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

– the title of the activity considered;

– a verdict and a supporting rationale for each assurance component that constitutes this activity, as a result of performing the corresponding CEM action and its constituent work units.

133 The rationale justifies the verdict using the CC, the CEM, any interpretations and the evaluation evidence examined and shows how the evaluation evidence does or does not meet each aspect of the criteria. It contains a description of the work performed, the method used, and any derivation of results. The rationale may provide detail to the level of a CEM work unit.

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

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

8.5.5.3.5 Conclusions and recommendations

136 The evaluator ***shall report*** the conclusions of the evaluation, which will relate to whether the TOE has satisfied its associated ST, in particular the overall verdict as defined in CC Part 1 Chapter 9, Evaluation results, and determined by application of the verdict assignment described in 8.2.5.

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

8.5.5.3.6 List of evaluation evidence

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

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

– the title;

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

8.5.5.3.7 List of acronyms/Glossary of terms

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

140 Glossary definitions already defined by the CC or CEM need not be repeated in the ETR.

8.5.5.3.8 Observation reports

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

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

9 Class APE: Protection Profile evaluation

9.1 Introduction

143 This Chapter describes the evaluation of a PP. The requirements and methodology for PP evaluation are identical for each PP evaluation, regardless of the EAL (or other set of assurance requirements) that is claimed in the PP. The evaluation methodology in this Chapter is based on the requirements on the PP as specified in CC Part 3 class APE.

144 This Chapter should be used in conjunction with Annexes A, B and C in CC Part 1, as these Annexes clarify the concepts here and provide many examples.

9.2 Application notes

9.2.1 Re-using the evaluation results of certified PPs

145 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, assumptions, 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.

146 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.

147 An example would be where the PP that conformance is being claimed to contains 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 ST 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.

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

9.3 PP introduction (APE_INT)

9.3.1 Evaluation of sub-activity (APE_INT.1)

9.3.1.1 Objectives

149 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.

9.3.1.2 Input

150 The evaluation evidence for this sub-activity is:

a) the PP.

9.3.1.3 Action APE_INT.1.1E

APE_INT.1.1C ***The PP introduction shall contain a PP reference and a TOE overview.***

APE_INT.1-1 The evaluator ***shall check*** that the PP introduction contains a PP reference and a TOE overview.

APE_INT.1.2C ***The PP reference shall uniquely identify the PP.***

APE_INT.1-2 The evaluator ***shall examine*** the PP reference to determine that it uniquely identifies the PP.

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

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

APE_INT.1.3C ***The TOE overview shall summarise the usage and major security features of the TOE.***

APE_INT.1-3 The evaluator ***shall examine*** the TOE overview to determine that it describes the usage and major security features of the TOE.

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

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

APE_INT.1.4C ***The TOE overview shall identify the TOE type.***

APE_INT.1-4 The evaluator ***shall check*** that the TOE overview identifies the TOE type.

APE_INT.1.5C ***The TOE overview shall identify any non-TOE hardware/software/firmware available to the TOE.***

APE_INT.1-5 The evaluator ***shall examine*** the TOE overview to determine that it identifies any non-TOE hardware/software/firmware available to the TOE.

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

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

9.4 Conformance claims (APE_CCL)

9.4.1 Evaluation of sub-activity (APE_CCL.1)

9.4.1.1 Objectives

157 The objective of this sub-activity is to determine the validity of various conformance claims. These describe how the PP conforms to the CC, other PPs and packages.

9.4.1.2 Input

158 The evaluation evidence for this sub-activity is:

a) the PP;

b) the PP(s) that the PP claims conformance to;

c) the package(s) that the PP claims conformance to.

9.4.1.3 Action APE_CCL.1.1E

APE_CCL.1.1C ***The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the PP claims conformance.***

APE_CCL.1-1 The evaluator ***shall check*** that the conformance claim contains a CC conformance claim that identifies the version of the CC to which the PP claims conformance.

159 The evaluator determines that the CC conformance claim identifies the version of the CC that was used to develop this PP. This should include the version number of the CC and, unless the International English version of the CC was used, the language of the version of the CC that was used.

APE_CCL.1.2C ***The CC conformance claim shall describe the conformance of the PP to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.***

APE_CCL.1-2 The evaluator ***shall check*** that the CC conformance claim states a claim of either CC Part 2 conformant or CC Part 2 extended for the PP.

APE_CCL.1.3C ***The CC conformance claim shall describe the conformance of the PP to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.***

APE_CCL.1-3 The evaluator ***shall check*** that the CC conformance claim states a claim of either CC Part 3 conformant or CC Part 3 extended for the PP.

APE_CCL.1.4C ***The CC conformance claim shall be consistent with the extended components definition.***

APE_CCL.1-4 The evaluator ***shall examine*** the CC conformance claim for CC Part 2 to determine that it is consistent with the extended components definition.

160 If the CC conformance claim contains CC Part 2 conformant, the evaluator determines that the extended components definition does not define functional components.

161 If the CC conformance claim contains CC Part 2 extended, the evaluator determines that the extended components definition defines at least one extended functional component.

APE_CCL.1-5 The evaluator ***shall examine*** the CC conformance claim for CC Part 3 to determine that it is consistent with the extended components definition.

162 If the CC conformance claim contains CC Part 3 conformant, the evaluator determines that the extended components definition does not define assurance components.

163 If the CC conformance claim contains CC Part 3 extended, the evaluator determines that the extended components definition defines at least one extended assurance component.

APE_CCL.1.5C ***The conformance claim shall identify all PPs and security requirement packages to which the PP claims conformance.***

APE_CCL.1-6 The evaluator ***shall check*** that the conformance claim contains a PP claim that identifies all PPs for which the PP claims conformance.

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

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

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

APE_CCL.1-7 The evaluator ***shall check*** that the conformance claim contains a package claim that identifies all packages to which the PP claims conformance.

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

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

169 The evaluator is reminded that claims of partial conformance to a package are not permitted.

APE_CCL.1.6C ***The conformance claim shall describe any conformance of the PP to a package as either package-conformant or package-augmented.***

APE_CCL.1-8 The evaluator ***shall check*** that, for each identified package, the conformance claim states a claim of either package-name conformant or package-name augmented.

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

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

a) If the package is an assurance package, then the PP contains all SARs included in the package, but no additional SARs.

b) If the package is a functional package, then the PP contains all SFRs included in the package, but no additional SFRs.

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

a) If the package is an assurance package, then the PP contains all SARs included in the package, and at least one additional SAR or at least one SAR that is hierarchical to a SAR in the package.

b) If the package is a functional package, then the PP contains all SFRs included in the package, and at least one additional SFR or at least one SFR that is hierarchical to a SFR in the package.

APE_CCL.1.7C ***The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.***

APE_CCL.1-9 The evaluator ***shall examine*** the conformance claim rationale to determine that the TOE type of the TOE is consistent with all TOE types of the PPs.

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

174 The relation between the types may be simple: a firewall PP claiming conformance to another firewall PP, or more complex: a smart card PP claiming conformance to a number of other PPs at the same time: a PP for the integrated circuit, a PP for the smart card OS, and two PPs for two applications on the smart card.

APE_CCL.1.8C ***The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.***

APE_CCL.1-10 The evaluator ***shall examine*** the conformance claim rationale to determine that it demonstrates that the statement of security problem definition is consistent, as defined by the conformance statement of the PP, with the

statements of security problem definition stated in the PPs to which conformance is being claimed.

175 If the PP under evaluation does not claim conformance with another PP, this work unit is not applicable and therefore considered to be satisfied.

176 If the PP to which conformance is being claimed does not have a statement of security problem definition, this work unit is not applicable and therefore considered to be satisfied.

177 If strict conformance is required by the PP to which conformance is being claimed, no conformance claim rationale is required. Instead, the evaluator determines whether

a) the threats in the PP under evaluation are a superset of or identical to the threats in the PP to which conformance is being claimed;

b) the OSPs in the PP under evaluation are a superset of or identical to the OSPs in the PP to which conformance is being claimed;

c) the assumptions in the PP under evaluation are identical to the OSPs in the PP to which conformance is being claimed;

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

179 For guidance on “equivalent or more restrictive” see CC Part 1 Annex D, PP conformance.

APE_CCL.1.9C ***The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.***

APE_CCL.1-11 The evaluator ***shall examine*** the conformance claim rationale to determine that the statement of security objectives is consistent, as defined by the conformance statement of the PPs, with the statement of security objectives in the PPs.

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

181 If strict conformance is required by the PP to which conformance is being claimed, no conformance claim rationale is required. Instead, the evaluator determines whether:

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

– The PP under evaluation contains exactly all security objectives for the operational environment (with one exception in the next bullet). Note that it is not allowed for the PP under evaluation to have additional security objectives for the operational environment;

– The PP under evaluation may specify that certain objectives for the operational environment in the PP that conformance is being claimed to are security objectives for the TOE in the PP under evaluation. This is a valid exception to the previous bullet.

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

183 For guidance on "equivalent or more restrictive" see CC Part 1 Annex D, PP conformance.

APE_CCL.1.10C ***The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.***

APE_CCL.1-12 The evaluator ***shall examine*** the PP to determine that it is consistent, as defined by the conformance statement of the PP, with all security requirements in the PPs for which conformance is being claimed.

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

185 If strict conformance is required by the PP to which conformance is being claimed, no conformance claim rationale is required. Instead, the evaluator determines whether the statement of security requirements in the PP under evaluation is a superset of or identical to the statement of security requirements in the PP to which conformance is being claimed (for strict conformance).

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

187 For guidance on "equivalent or more restrictive" see CC Part 1 Annex D, PP conformance.

APE_CCL.1.11C ***The conformance statement shall describe the conformance required of any PPs/STs to the PP as strict-PP or demonstrable-PP conformance.***

APE_CCL.1-13 The evaluator ***shall check*** that the PP conformance statement states a claim of strict-PP or demonstrable-PP conformance.

9.5 Security problem definition (APE_SPD)

9.5.1 Evaluation of sub-activity (APE_SPD.1)

9.5.1.1 Objectives

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

9.5.1.2 Input

189 The evaluation evidence for this sub-activity is:

a) the PP.

9.5.1.3 Action APE_SPD.1.1E

APE_SPD.1.1C ***The security problem definition shall describe the threats.***

APE_SPD.1-1 The evaluator ***shall check*** that the security problem definition describes the threats.

190 If all security objectives are derived from assumptions and/or OSPs only, the statement of threats need not be present in the PP. In this case, this work unit is not applicable and therefore considered to be satisfied.

191 The evaluator determines that the security problem definition describes the threats that must be countered by the TOE and/or its operational environment.

APE_SPD.1.2C ***All threats shall be described in terms of a threat agent, an asset, and an adverse action.***

APE_SPD.1-2 The evaluator ***shall examine*** the security problem definition to determine that all threats are described in terms of a threat agent, an asset, and an adverse action.

192 If all security objectives are derived from assumptions and OSPs only, the statement of threats need not be present in the PP. In this case, this work unit is not applicable and therefore considered to be satisfied.

193 Threat agents may be further described by aspects such as expertise, resource, opportunity, and motivation.

APE_SPD.1.3C ***The security problem definition shall describe the OSPs.***

APE_SPD.1-3 The evaluator ***shall check*** that the security problem definition describes the OSPs.

194 If all security objectives are derived from assumptions and/or threats only, OSPs need not be present in the PP. In this case, this work unit is not applicable and therefore considered to be satisfied.

195 The evaluator determines that OSP statements are made in terms of rules or guidelines that must be followed by the TOE and/or its operational environment.

196 The evaluator determines that each OSP is explained and/or interpreted in sufficient detail to make it clearly understandable; a clear presentation of policy statements is necessary to permit tracing security objectives to them.

APE_SPD.1.4C ***The security problem definition shall describe the assumptions about the operational environment of the TOE.***

APE_SPD.1-4 The evaluator ***shall examine*** the security problem definition to determine that it describes the assumptions about the operational environment of the TOE.

197 If there are no assumptions, this work unit is not applicable and is therefore considered to be satisfied.

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

9.6 Security objectives (APE_OBJ)

9.6.1 Evaluation of sub-activity (APE_OBJ.1)

9.6.1.1 Objectives

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

9.6.1.2 Input

200 The evaluation evidence for this sub-activity is:

a) the PP.

9.6.1.3 Action APE_OBJ.1.1E

APE_OBJ.1.1C ***The statement of security objectives shall describe the security objectives for the operational environment.***

APE_OBJ.1-1 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the operational environment.

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

9.6.2 Evaluation of sub-activity (APE_OBJ.2)

9.6.2.1 Objectives

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

9.6.2.2 Input

203 The evaluation evidence for this sub-activity is:

a) the PP.

9.6.2.3 Action APE_OBJ.2.1E

APE_OBJ.2.1C ***The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.***

APE_OBJ.2-1 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the TOE and the security objectives for the operational environment.

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

APE_OBJ.2.2C ***The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.***

APE_OBJ.2-2 The evaluator ***shall check*** that the security objectives rationale traces all security objectives for the TOE back to threats countered by the objectives and/or OSPs enforced by the objectives.

205 Each security objective for the TOE may trace back to threats or OSPs, or a combination of threats and OSPs, but it must trace back to at least one threat or OSP.

206 Failure to trace implies that either the security objectives rationale is incomplete, the security problem definition is incomplete, or the security objective for the TOE has no useful purpose.

APE_OBJ.2.3C ***The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.***

APE_OBJ.2-3 The evaluator ***shall check*** that the security objectives rationale traces the security objectives for the operational environment back to threats countered by that security objective, to OSPs enforced by that security objective, and to assumptions upheld by that security objective.

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

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

APE_OBJ.2.4C ***The security objectives rationale shall demonstrate that the security objectives counter all threats.***

APE_OBJ.2-4 The evaluator ***shall examine*** the security objectives rationale to determine that it justifies for each threat that the security objectives are suitable to counter that threat.

209 If no security objectives trace back to the threat, this work unit fails.

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

211 The evaluator determines that the justification for a threat demonstrates that the security objectives are sufficient: if all security objectives that trace back to the threat are achieved, the threat is removed, sufficiently diminished, or the effects of the threat are sufficiently mitigated.

212 Note that the tracings from security objectives to threats provided in the security objectives rationale may be part of a justification, but do not constitute a justification by themselves. Even in the case that a security objective is merely a statement reflecting the intent to prevent a particular threat from being realised, a justification is required, but this justification may be as minimal as "Security Objective X directly counters Threat Y".

213 The evaluator also determines that each security objective that traces back to a threat is necessary: when the security objective is achieved it actually contributes to the removal, diminishing or mitigation of that threat.

APE_OBJ.2.5C ***The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.***

APE_OBJ.2-5 The evaluator ***shall examine*** the security objectives rationale to determine that for each OSP it justifies that the security objectives are suitable to enforce that OSP.

214 If no security objectives trace back to the OSP, this work unit fails.

215 The evaluator determines that the justification for an OSP demonstrates that the security objectives are sufficient: if all security objectives that trace back to that OSP are achieved, the OSP is enforced.

216 The evaluator also determines that each security objective that traces back to an OSP is necessary: when the security objective is achieved it actually contributes to the enforcement of the OSP.

217 Note that the tracings from security objectives to OSPs provided in the security objectives rationale may be part of a justification, but do not constitute a justification by themselves. In the case that a security objective is merely a statement reflecting the intent to enforce a particular OSP, a justification is required, but this justification may be as minimal as "Security Objective X directly enforces OSP Y".

APE_OBJ.2.6C ***The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.***

APE_OBJ.2-6 The evaluator ***shall examine*** the security objectives rationale to determine that for each assumption for the operational environment it contains an appropriate justification that the security objectives for the operational environment are suitable to uphold that assumption.

218 If no security objectives for the operational environment trace back to the assumption, this work unit fails.

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

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

221 Note that the tracings from security objectives for the operational environment to assumptions provided in the security objectives rationale may be a part of a justification, but do not constitute a justification by themselves. Even in the case that a security objective of the operational environment is merely a restatement of an assumption, a justification is required, but this justification may be as minimal as “Security Objective X directly upholds Assumption Y”.

9.7 Extended components definition (APE_ECD)

9.7.1 Evaluation of sub-activity (APE_ECD.1)

9.7.1.1 Objectives

222 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 CC Part 2 or CC Part 3 components.

9.7.1.2 Input

223 The evaluation evidence for this sub-activity is:

a) the PP.

9.7.1.3 Action APE_ECD.1.1E

APE_ECD.1.1C ***The statement of security requirements shall identify all extended security requirements.***

APE_ECD.1-1 The evaluator ***shall check*** that all security requirements in the statement of security requirements that are not identified as extended requirements are present in CC Part 2 or in CC Part 3.

APE_ECD.1.2C ***The extended components definition shall define an extended component for each extended security requirement.***

APE_ECD.1-2 The evaluator ***shall check*** that the extended components definition defines an extended component for each extended security requirement.

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

225 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.

APE_ECD.1.3C ***The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.***

APE_ECD.1-3 The evaluator ***shall examine*** the extended components definition to determine that it describes how each extended component fits into the existing CC components, families, and classes.

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

227 The evaluator determines that each extended component is either:

a) a member of an existing CC Part 2 or CC Part 3 family, or

b) a member of a new family defined in the PP.

228 If the extended component is a member of an existing CC Part 2 or CC Part 3 family, the evaluator determines that the extended components definition adequately describes why the extended component should be a member of that family and how it relates to other components of that family.

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

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

a) a member of an existing CC Part 2 or CC Part 3 class, or

b) a member of a new class defined in the PP.

231 If the family is a member of an existing CC Part 2 or CC Part 3 class, the evaluator determines that the extended components definition adequately describes why the family should be a member of that class and how it relates to other families in that class.

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

APE_ECD.1-4 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended component identifies all applicable dependencies of that component.

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

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

APE_ECD.1.4C ***The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.***

APE_ECD.1-5 The evaluator ***shall examine*** the extended components definition to determine that each extended functional component uses the existing CC Part 2 components as a model for presentation.

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

236 The evaluator determines that the extended functional component is consistent with CC Part 2 Section 7.1.3, Component structure.

237 If the extended functional component uses operations, the evaluator determines that the extended functional component is consistent with CC Part 1 Annex C.4, Operations.

238 If the extended functional component is hierarchical to an existing functional component, the evaluator determines that the extended functional component is consistent with CC Part 2 Section 7.2.1, Component changes highlighting.

APE_ECD.1-6 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new functional family uses the existing CC functional families as a model for presentation.

239 If the PP does not define new functional families, this work unit is not applicable and therefore considered to be satisfied.

240 The evaluator determines that all new functional families are defined consistent with CC Part 2 Section 7.1.2, Family structure.

APE_ECD.1-7 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new functional class uses the existing CC functional classes as a model for presentation.

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

242 The evaluator determines that all new functional classes are defined consistent with CC Part 2 Section 7.1.1, Class structure

APE_ECD.1-8 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended assurance component uses the existing CC Part 3 components as a model for presentation.

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

244 The evaluator determines that the extended assurance component definition is consistent with CC Part 3 Section 7.1.3, Assurance component structure.

245 If the extended assurance component uses operations, the evaluator determines that the extended assurance component is consistent with CC Part 1 Annex C.4, Operations.

246 If the extended assurance component is hierarchical to an existing assurance component, the evaluator determines that the extended assurance component is consistent with CC Part 3 Section 7.1.3, Assurance component structure.

APE_ECD.1-9 The evaluator ***shall examine*** the extended components definition to determine that, for each defined extended assurance component, applicable methodology has been provided.

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

248 The evaluator determines that, for each evaluator action element of each extended SAR, one or more work units are provided and that successfully

performing all work units for a given evaluator action element will demonstrate that the element has been achieved.

APE_ECD.1-10 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new assurance family uses the existing CC assurance families as a model for presentation.

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

250 The evaluator determines that all new assurance families are defined consistent with CC Part 3 Section 7.1.2, Assurance family structure.

APE_ECD.1-11 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new assurance class uses the existing CC assurance classes as a model for presentation.

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

252 The evaluator determines that all new assurance classes are defined consistent with CC Part 3 Section 7.1.1, Assurance class structure.

APE_ECD.1.5C ***The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.***

APE_ECD.1-12 The evaluator ***shall examine*** the extended components definition to determine that each element in each extended component is measurable and states objective evaluation requirements, such that conformance or nonconformance can be demonstrated.

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

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

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

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

9.7.1.4 Action APE_ECD.1.2E

APE_ECD.1-13 The evaluator ***shall examine*** the extended components definition to determine that each extended component may not be clearly expressed using existing components.

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

258 The evaluator should take components from CC Part 2 and CC Part 3, other extended components that have been defined in the PP, combinations of these components, and possible operations on these components into account when making this determination.

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

9.8 Security requirements (APE_REQ)

9.8.1 Evaluation of sub-activity (APE_REQ.1)

9.8.1.1 Objectives

260 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and well-defined and whether they are internally consistent.

9.8.1.2 Input

261 The evaluation evidence for this sub-activity is:

a) the PP.

9.8.1.3 Action APE_REQ.1.1E

APE_REQ.1.1C ***The statement of security requirements shall describe the SFRs and the SARs.***

APE_REQ.1-1 The evaluator ***shall check*** that the statement of security requirements describes the SFRs.

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

a) by reference to an individual component in CC Part 2;

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

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

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

e) by reproduction in the PP.

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

APE_REQ.1-2 The evaluator ***shall check*** that the statement of security requirements describes the SARs.

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

a) by reference to an individual component in CC Part 3;

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

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

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

e) by reproduction in the PP.

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

APE_REQ.1.2C ***All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.***

APE_REQ.1-3 The evaluator ***shall examine*** the PP to determine that all subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

266 The evaluator determines that the PP defines all:

- (types of) subjects and objects that are used in the SFRs;
- (types of) security attributes of subjects, users, objects, information, sessions and/or resources, possible values that these attributes may take and any relations between these values (e.g. top_secret is "higher" than secret);
- (types of) operations that are used in the SFRs, including the effects of these operations;
- (types of) external entities in the SFRs;
- other terms that are introduced in the SFRs and/or SARs by completing operations, if these terms are not immediately clear, or are used outside their dictionary definition.

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

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

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

APE_REQ.1.3C ***The statement of security requirements shall identify all operations on the security requirements.***

APE_REQ.1-4 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

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

APE_REQ.1.4C ***All operations shall be performed correctly.***

APE_REQ.1-5 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

271 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.1-6 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration operations are performed correctly.

272 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.1-7 The evaluator ***shall examine*** the statement of security requirements to determine that all selection operations are performed correctly.

273 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.1-8 The evaluator ***shall examine*** the statement of security requirements to determine that all refinement operations are performed correctly.

274 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.1.5C ***Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.***

APE_REQ.1-9 The evaluator ***shall examine*** the statement of security requirements to determine that each dependency of the security requirements is either satisfied, or that the security requirements rationale justifies the dependency not being satisfied.

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

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

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

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

APE_REQ.1.6C ***The statement of security requirements shall be internally consistent.***

APE_REQ.1-10 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

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

279 Some possible conflicts are:

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

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

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

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

9.8.2 Evaluation of sub-activity (APE_REQ.2)

9.8.2.1 Objectives

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

9.8.2.2 Input

281 The evaluation evidence for this sub-activity is:

a) the PP.

9.8.2.3 Action APE_REQ.2.1E

APE_REQ.2.1C ***The statement of security requirements shall describe the SFRs and the SARs.***

APE_REQ.2-1 The evaluator ***shall check*** that the statement of security requirements describes the SFRs.

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

a) by reference to an individual component in CC Part 2;

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

c) by reference to an individual component in a PP that the PP claims to be conformant with;

d) by reference to an individual component in a security requirements package that the PP claims to be conformant with;

e) by reproduction in the PP.

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

APE_REQ.2-2 The evaluator ***shall check*** that the statement of security requirements describes the SARs.

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

a) by reference to an individual component in CC Part 3;

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

c) by reference to an individual component in a PP that the PP claims to be conformant with;

d) by reference to an individual component in a security requirements package that the PP claims to be conformant with;

e) by reproduction in the PP.

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

APE_REQ.2.2C ***All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.***

APE_REQ.2-3 The evaluator ***shall examine*** the PP to determine that all subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

286 The evaluator determines that the PP defines all:

- (types of) subjects and objects that are used in the SFRs;
- (types of) security attributes of subjects, users, objects, information, sessions and/or resources, possible values that these attributes may take and any relations between these values (e.g. top_secret is "higher" than secret);
- (types of) operations that are used in the SFRs, including the effects of these operations;
- (types of) external entities in the SFRs;
- other terms that are introduced in the SFRs and/or SARs by completing operations, if these terms are not immediately clear, or are used outside their dictionary definition.

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

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

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

APE_REQ.2.3C ***The statement of security requirements shall identify all operations on the security requirements.***

APE_REQ.2-4 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

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

APE_REQ.2.4C ***All operations shall be performed correctly.***

APE_REQ.2-5 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

291 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.2-6 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration operations are performed correctly.

292 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.2-7 The evaluator ***shall examine*** the statement of security requirements to determine that all selection operations are performed correctly.

293 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.2-8 The evaluator ***shall examine*** the statement of security requirements to determine that all refinement operations are performed correctly.

294 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

APE_REQ.2.5C ***Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.***

APE_REQ.2-9 The evaluator ***shall examine*** the statement of security requirements to determine that each dependency of the security requirements is either satisfied, or that the security requirements rationale justifies the dependency not being satisfied.

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

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

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

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

APE_REQ.2.6C ***The security requirements rationale shall trace each SFR back to the security objectives for the TOE.***

APE_REQ.2-10 The evaluator ***shall check*** that the security requirements rationale traces each SFR back to the security objectives for the TOE.

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

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

APE_REQ.2.7C ***The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.***

APE_REQ.2-11 The evaluator ***shall examine*** the security requirements rationale to determine that for each security objective for the TOE it justifies that the SFRs are suitable to meet that security objective for the TOE.

299 If no SFRs trace back to the security objective for the TOE, this work unit fails.

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

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

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

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

APE_REQ.2.8C ***The security requirements rationale shall explain why the SARs were chosen.***

APE_REQ.2-12 The evaluator ***shall check*** that the security requirements rationale explains why the SARs were chosen.

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

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

APE_REQ.2.9C ***The statement of security requirements shall be internally consistent.***

APE_REQ.2-13 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

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

308 Some possible conflicts are:

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

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

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

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

10 Class ASE: Security Target evaluation

10.1 Introduction

309 This Chapter describes the evaluation of an ST. The ST evaluation should be started prior to any TOE evaluation sub-activities since the ST provides the basis and context to perform these sub-activities. The evaluation methodology in this section is based on the requirements on the ST as specified in CC Part 3 class ASE.

310 This Chapter should be used in conjunction with Annexes A, B and C in CC Part 1, as these Annexes clarify the concepts here and provide many examples.

10.2 Application notes

10.2.1 Re-using the evaluation results of certified PPs

311 While evaluating an ST 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 ST does not add threats, OSPs, assumptions, security objectives and/or security requirements to those of the PP. If the ST contains much more than the certified PP, re-use may not be useful at all.

312 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.

313 An example would be where the PP contains a set of security requirements, and these were determined to be internally consistent during the PP evaluation. If the ST uses the exact same requirements, the consistency analysis does not have to be repeated during the ST evaluation. If the ST 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.

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

10.3 ST introduction (ASE_INT)

10.3.1 Evaluation of sub-activity (ASE_INT.1)

10.3.1.1 Objectives

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

10.3.1.2 Input

316 The evaluation evidence for this sub-activity is:

a) the ST.

10.3.1.3 Action ASE_INT.1.1E

ASE_INT.1.1C ***The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.***

ASE_INT.1-1 The evaluator ***shall check*** that the ST introduction contains an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C ***The ST reference shall uniquely identify the ST.***

ASE_INT.1-2 The evaluator ***shall examine*** the ST reference to determine that it uniquely identifies the ST.

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

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

ASE_INT.1.3C ***The TOE reference shall identify the TOE.***

ASE_INT.1-3 The evaluator ***shall examine*** the TOE reference to determine that it identifies the TOE.

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

ASE_INT.1-4 The evaluator ***shall examine*** the TOE reference to determine that it is not misleading.

320 If the TOE is related to one or more well-known products, it is allowed to reflect this in the TOE reference. However, this should not be used to mislead consumers: situations where only a small part of a product is evaluated, yet the TOE reference does not reflect this, are not allowed.

ASE_INT.1.4C ***The TOE overview shall summarise the usage and major security features of the TOE.***

ASE_INT.1-5 The evaluator ***shall examine*** the TOE overview to determine that it describes the usage and major security features of the TOE.

321 The TOE overview should briefly (i.e. several paragraphs) describe the usage and major security features of the TOE. The TOE overview should enable potential consumers to quickly determine whether the TOE may be suitable for their security needs.

322 The TOE overview in an ST for a composed TOE should describe the usage and major security feature of the composed TOE, rather than those of the individual component TOEs.

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

ASE_INT.1.5C ***The TOE overview shall identify the TOE type.***

ASE_INT.1-6 The evaluator ***shall check*** that the TOE overview identifies the TOE type.

ASE_INT.1-7 The evaluator ***shall examine*** the TOE overview to determine that the TOE type is not misleading.

324 There are situations where the general consumer would expect certain functionality of the TOE because of its TOE type. If this functionality is absent in the TOE, the evaluator determines that the TOE overview adequately discusses this absence.

325 There are also TOEs where the general consumer would expect that the TOE should be able to operate in a certain operational environment because of its TOE type. If the TOE is unable to operate in such an operational environment, the evaluator determines that the TOE overview adequately discusses this.

ASE_INT.1.6C ***The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.***

ASE_INT.1-8 The evaluator ***shall examine*** the TOE overview to determine that it identifies any non-TOE hardware/software/firmware required by the TOE.

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

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

ASE_INT.1.7C ***The TOE description shall describe the physical scope of the TOE.***

ASE_INT.1-9 The evaluator ***shall examine*** the TOE description to determine that it describes the physical scope of the TOE.

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

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

ASE_INT.1.8C ***The TOE description shall describe the logical scope of the TOE.***

ASE_INT.1-10 The evaluator ***shall examine*** the TOE description to determine that it describes the logical scope of the TOE.

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

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

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

10.3.1.4 Action ASE_INT.1.2E

ASE_INT.1-11 The evaluator ***shall examine*** the TOE reference, TOE overview and TOE description to determine that they are consistent with each other.

10.4 Conformance claims (ASE_CCL)

10.4.1 Evaluation of sub-activity (ASE_CCL.1)

10.4.1.1 Objectives

333 The objective of this sub-activity is to determine the validity of various conformance claims. These describe how the ST and the TOE conform to the CC and how the ST conforms to PPs and packages.

10.4.1.2 Input

334 The evaluation evidence for this sub-activity is:

a) the ST;

b) the PP(s) that the ST claims conformance to;

c) the package(s) that the ST claims conformance to.

10.4.1.3 Action ASE_CCL.1.1E

ASE_CCL.1.1C ***The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.***

ASE_CCL.1-1 The evaluator ***shall check*** that the conformance claim contains a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

335 The evaluator determines that the CC conformance claim identifies the version of the CC that was used to develop this ST. This should include the version number of the CC and, unless the International English version of the CC was used, the language of the version of the CC that was used.

336 For a composed TOE, the evaluator will consider any differences between the version of the CC claimed for a component and the version of the CC claimed for the composed TOE. If the versions differ the evaluator will assess whether the differences between the versions will lead to conflicting claims.

337 For instances where the CC conformance claims for the base TOE and dependent TOE are for different major releases of the CC (e.g. one component TOE conformance claim is CC v2.x and the other component TOE conformance claim is CC v3.x), the conformance claim for the composed TOE will be the earlier release of the CC, as the CC is developed with an aim to provide backwards compatibility (although this may not be achieved in the strictest sense, it is understood to be achieved in principle).

ASE_CCL.1.2C ***The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.***

ASE_CCL.1-2 The evaluator ***shall check*** that the CC conformance claim states a claim of either CC Part 2 conformant or CC Part 2 extended for the ST.

338 For a composed TOE, the evaluator will consider whether this claim is consistent not only with the CC Part 2, but also with the claims of conformance to CC Part 2 by each of the component TOEs. I.e. if one or more component TOEs claims to be CC Part 2 extended, then the composed TOE should also claim to be CC Part 2 extended.

339 The CC conformance claim for the composed TOE may be CC Part 2 extended, even though the component TOEs are Part 2 conformant, in the event that additional SFRs are claimed for the base TOE (see composed TOE guidance for ASE_CCL.1.6C)

ASE_CCL.1.3C ***The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.***

ASE_CCL.1-3 The evaluator ***shall check*** that the CC conformance claim states a claim of either CC Part 3 conformant or CC Part 3 extended for the ST.

ASE_CCL.1.4C ***The CC conformance claim shall be consistent with the extended components definition.***

ASE_CCL.1-4 The evaluator ***shall examine*** the CC conformance claim for CC Part 2 to determine that it is consistent with the extended components definition.

340 If the CC conformance claim contains CC Part 2 conformant, the evaluator determines that the extended components definition does not define functional components.

341 If the CC conformance claim contains CC Part 2 extended, the evaluator determines that the extended components definition defines at least one extended functional component.

ASE_CCL.1-5 The evaluator ***shall examine*** the CC conformance claim for CC Part 3 to determine that it is consistent with the extended components definition.

342 If the CC conformance claim contains CC Part 3 conformant, the evaluator determines that the extended components definition does not define assurance components.

343 If the CC conformance claim contains CC Part 3 extended, the evaluator determines that the extended components definition defines at least one extended assurance component.

ASE_CCL.1.5C ***The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.***

ASE_CCL.1-6 The evaluator ***shall check*** that the conformance claim contains a PP claim that identifies all PPs for which the ST claims conformance.

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

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

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

347 The ST for a composed TOE will identify the STs of the component TOEs from which the composed ST is comprised. The composed TOE is essentially claiming conformance to the STs of the component TOEs.

ASE_CCL.1-7 The evaluator ***shall check*** that the conformance claim contains a package claim that identifies all packages to which the ST claims conformance.

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

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

350 The evaluator determines that the component TOE STs from which the composed TOE is derived are also unambiguously identified.

351 The evaluator is reminded that claims of partial conformance to a package are not permitted.

ASE_CCL.1.6C ***The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.***

ASE_CCL.1-8 The evaluator ***shall check*** that, for each identified package, the conformance claim states a claim of either package-name conformant or package-name augmented.

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

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

a) If the package is an assurance package, then the ST contains all SARs included in the package, but no additional SARs.

b) If the package is a functional package, then the ST contains all SFRs included in the package, but no additional SFRs.

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

a) If the package is an assurance package then the ST contains all SARs included in the package, and at least one additional SAR or at least one SAR that is hierarchical to a SAR in the package.

b) If the package is a functional package, then the ST contains all SFRs included in the package, and at least one additional SFR or at least one SFR that is hierarchical to a SFR in the package.

ASE_CCL.1.7C ***The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.***

ASE_CCL.1-9 The evaluator ***shall examine*** the conformance claim rationale to determine that the TOE type of the TOE is consistent with all TOE types of the PPs.

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

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

357 For a composed TOE, the evaluator will determine whether the conformance claim rationale demonstrates that the TOE types of the component TOEs are consistent with the composed TOE type. This does not mean that both the component and the composed TOE types have to be the same, but rather that the component TOEs are suitable for integration to provide the composed TOE. It should be made clear in the composed TOE ST which SFRs are only included as a result of composition, and were not examined as SFRs in the base and dependent TOE (e.g. EALx) evaluation.

ASE_CCL.1.8C ***The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.***

ASE_CCL.1-10 The evaluator ***shall examine*** the conformance claim rationale to determine that it demonstrates that the statement of security problem definition is consistent, as defined by the conformance statement of the PP, with the statements of security problem definition stated in the PPs to which conformance is being claimed.

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

359 If the PP does not have a statement of security problem definition, this work unit is not applicable and therefore considered to be satisfied.

360 If strict conformance is required by the PP to which conformance is being claimed no conformance claim rationale is required. Instead, the evaluator determines whether:

a) the threats in the ST are a superset of or identical to the threats in the PP to which conformance is being claimed;

b) the OSPs in the ST are a superset of or identical to the OSPs in the PP to which conformance is being claimed;

c) the assumptions in the ST are identical to the assumptions in the PP to which conformance is being claimed;

361 If demonstrable conformance is required by the PP, the evaluator examines the conformance claim rationale to determine that it demonstrates that the statement of security problem definition of the ST is equivalent or more restrictive than the statement of security problem definition in the PP to which conformance is being claimed.

362 For guidance on “equivalent or more restrictive” see CC Part 1 Annex D, PP conformance.

363 For a composed TOE, the evaluator will consider whether the security problem definition of the composed TOE is consistent with that specified in the STs for the component TOEs. This is determined in terms of demonstrable conformance. In particular, the evaluator examines the conformance claim rationale to determine that:

a) Threat statements and OSPs in the composed TOE ST do not contradict those from the component STs.

b) Any assumptions made in the component STs are upheld in the composed TOE ST. That is, either the assumption should also be present in the composed ST, or the assumption should be positively addressed in the composed ST. The assumption may be positively addressed through specification of requirements in the composed TOE to provide functionality fulfilling the concern captured in the assumption.

ASE_CCL.1.9C ***The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.***

ASE_CCL.1-11 The evaluator ***shall examine*** the conformance claim rationale to determine that the statement of security objectives is consistent, as defined by the

conformance statement of the PP, with the statement of security objectives in the PPs.

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

365 If strict conformance is required by the PP, no conformance claim rationale is required. Instead, the evaluator determines whether:

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

– The ST contains exactly all security objectives for the operational environment (with one exception in the next bullet). Note that it is not allowed for the ST under evaluation to have additional security objectives for the operational environment;

– The ST may specify that certain objectives for the operational environment in the PP that conformance is being claimed to are security objectives for the TOE in the ST. This is a valid exception to the previous bullet.

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

367 For guidance on “equivalent or more restrictive” see CC Part 1 Annex D, PP conformance.

368 For a composed TOE, the evaluator will consider whether the security objectives of the composed TOE are consistent with that specified in the STs for the component TOEs. This is determined in terms of demonstrable conformance. In particular, the evaluator examines the conformance claim rationale to determine that:

a) The statement of security objectives in the dependent TOE ST relevant to any IT in the operational environment are consistent with the statement of security objectives for the TOE in the base TOE ST. It is not expected that the statement of security objectives for the environment within in the dependent TOE ST will cover all aspects of the statement of security objectives for the TOE in the base TOE ST.

b) The statement of security objectives in the composed ST is consistent with the statements of security objectives in the STs for the component TOEs.

369 If demonstrable conformance is required by the PP, the evaluator examines the conformance claim rationale to determine that it demonstrates that the statement of security objectives of the ST is at least equivalent to the statement of security objectives in the PP, or component TOE ST in the case of a composed TOE ST.

ASE_CCL.1.10C ***The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.***

ASE_CCL.1-12 The evaluator ***shall examine*** the ST to determine that it is consistent, as defined by the conformance statement of the PP, with all security requirements in the PPs for which conformance is being claimed.

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

371 If strict conformance is required by the PP to which conformance is being claimed, no conformance claim rationale is required. Instead, the evaluator determines whether the statement of security requirements in the ST is a superset of or identical to the statement of security requirements in the PP to which conformance is being claimed (for strict conformance).

372 If demonstrable conformance is required by the PP to which conformance is being claimed, the evaluator examines the conformance claim rationale to determine that it demonstrates that the statement of security requirements of the ST is equivalent or more restrictive than the statement of security requirements in the PP to which conformance is being claimed.

373 For guidance on “equivalent or more restrictive” see CC Part 1 Annex D, PP conformance.

374 For a composed TOE, the evaluator will consider whether the security requirements of the composed TOE are consistent with that specified in the STs for the component TOEs. This is determined in terms of demonstrable conformance. In particular, the evaluator examines the conformance rationale to determine that:

a) The statement of security requirements in the dependent TOE ST relevant to any IT in the operational environment is consistent with the statement of security requirements for the TOE in the base TOE ST. It is not expected that the statement of security requirements for the environment within in the dependent TOE ST will cover all aspects of the statement of security requirements for the TOE in the base TOE ST, as some SFRs may need to be added to the statement of security requirements in the composed TOE ST. However, the statement of security requirements in the base should support the operation of the dependent component.

b) The statement of security objectives in the dependent TOE ST relevant to any IT in the operational environment is consistent with

the statement of security requirements for the TOE in the base TOE ST. It is not expected that the statement of security objectives for the environment within in the dependent TOE ST will cover all aspects of the statement of security requirements for the TOE in the base TOE ST.

c) The statement of security requirements in the composed is consistent with the statements of security requirements in the STs for the component TOEs.

375 If demonstrable conformance is required by the PP to which conformance is being claimed, the evaluator examines the conformance claim rationale to determine that it demonstrates that the statement of security requirements of the ST is at least equivalent to the statement of security requirements in the PP, or component TOE ST in the case of a composed TOE ST.

10.5 Security problem definition (ASE_SPD)

10.5.1 Evaluation of sub-activity (ASE_SPD.1)

10.5.1.1 Objectives

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

10.5.1.2 Input

377 The evaluation evidence for this sub-activity is:

a) the ST.

10.5.1.3 Action ASE_SPD.1.1E

ASE_SPD.1.1C ***The security problem definition shall describe the threats.***

ASE_SPD.1-1 The evaluator ***shall check*** that the security problem definition describes the threats.

378 If all security objectives are derived from assumptions and/or OSPs only, the statement of threats need not be present in the ST. In this case, this work unit is not applicable and therefore considered to be satisfied.

379 The evaluator determines that the security problem definition describes the threats that must be countered by the TOE and/or operational environment.

ASE_SPD.1.2C ***All threats shall be described in terms of a threat agent, an asset, and an adverse action.***

ASE_SPD.1-2 The evaluator ***shall examine*** the security problem definition to determine that all threats are described in terms of a threat agent, an asset, and an adverse action.

380 If all security objectives are derived from assumptions and/or OSPs only, the statement of threats need not be present in the ST. In this case, this work unit is not applicable and therefore considered to be satisfied.

381 Threat agents may be further described by aspects such as expertise, resource, opportunity, and motivation.

ASE_SPD.1.3C ***The security problem definition shall describe the OSPs.***

ASE_SPD.1-3 The evaluator ***shall check*** that the security problem definition describes the OSPs.

382 If all security objectives are derived from assumptions and threats only, OSPs need not be present in the ST. In this case, this work unit is not applicable and therefore considered to be satisfied.

383 The evaluator determines that OSP statements are made in terms of rules or guidelines that must be followed by the TOE and/or its operational environment.

384 The evaluator determines that each OSP is explained and/or interpreted in sufficient detail to make it clearly understandable; a clear presentation of policy statements is necessary to permit tracing security objectives to them.

ASE_SPD.1.4C ***The security problem definition shall describe the assumptions about the operational environment of the TOE.***

ASE_SPD.1-4 The evaluator ***shall examine*** the security problem definition to determine that it describes the assumptions about the operational environment of the TOE.

385 If there are no assumptions, this work unit is not applicable and is therefore considered to be satisfied.

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

10.6 Security objectives (ASE_OBJ)

10.6.1 Evaluation of sub-activity (ASE_OBJ.1)

10.6.1.1 Objectives

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

10.6.1.2 Input

388 The evaluation evidence for this sub-activity is:

a) the ST.

10.6.1.3 Action ASE_OBJ.1.1E

ASE_OBJ.1.1C ***The statement of security objectives shall describe the security objectives for the operational environment.***

ASE_OBJ.1-1 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the operational environment.

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

10.6.2 Evaluation of sub-activity (ASE_OBJ.2)

10.6.2.1 Objectives

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

10.6.2.2 Input

391 The evaluation evidence for this sub-activity is:

a) the ST.

10.6.2.3 Action ASE_OBJ.2.1E

ASE_OBJ.2.1C ***The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.***

ASE_OBJ.2-1 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the TOE and the security objectives for the operational environment.

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

ASE_OBJ.2.2C ***The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.***

ASE_OBJ.2-2 The evaluator ***shall check*** that the security objectives rationale traces all security objectives for the TOE back to threats countered by the objectives and/or OSPs enforced by the objectives.

393 Each security objective for the TOE may trace back to threats or OSPs, or a combination of threats and OSPs, but it must trace back to at least one threat or OSP.

394 Failure to trace implies that either the security objectives rationale is incomplete, the security problem definition is incomplete, or the security objective for the TOE has no useful purpose.

ASE_OBJ.2.3C ***The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.***

ASE_OBJ.2-3 The evaluator ***shall check*** that the security objectives rationale traces the security objectives for the operational environment back to threats countered by that security objective, to OSPs enforced by that security objective, and to assumptions upheld by that security objective.

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

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

ASE_OBJ.2.4C ***The security objectives rationale shall demonstrate that the security objectives counter all threats.***

ASE_OBJ.2-4 The evaluator ***shall examine*** the security objectives rationale to determine that it justifies for each threat that the security objectives are suitable to counter that threat.

397 If no security objectives trace back to the threat, this work unit fails.

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

399 The evaluator determines that the justification for a threat demonstrates that the security objectives are sufficient: if all security objectives that trace back to the threat are achieved, the threat is removed, sufficiently diminished, or the effects of the threat are sufficiently mitigated.

400 Note that the tracings from security objectives to threats provided in the security objectives rationale may be part of a justification, but do not constitute a justification by themselves. Even in the case that a security objective is merely a statement reflecting the intent to prevent a particular threat from being realised, a justification is required, but this justification may be as minimal as "Security Objective X directly counters Threat Y".

401 The evaluator also determines that each security objective that traces back to a threat is necessary: when the security objective is achieved it actually contributes to the removal, diminishing or mitigation of that threat.

ASE_OBJ.2.5C ***The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.***

ASE_OBJ.2-5 The evaluator ***shall examine*** the security objectives rationale to determine that for each OSP it justifies that the security objectives are suitable to enforce that OSP.

402 If no security objectives trace back to the OSP, this work unit fails.

403 The evaluator determines that the justification for an OSP demonstrates that the security objectives are sufficient: if all security objectives that trace back to that OSP are achieved, the OSP is enforced.

404 The evaluator also determines that each security objective that traces back to an OSP is necessary: when the security objective is achieved it actually contributes to the enforcement of the OSP.

405 Note that the tracings from security objectives to OSPs provided in the security objectives rationale may be part of a justification, but do not constitute a justification by themselves. In the case that a security objective is merely a statement reflecting the intent to enforce a particular OSP, a justification is required, but this justification may be as minimal as "Security Objective X directly enforces OSP Y".

ASE_OBJ.2.6C ***The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.***

ASE_OBJ.2-6 The evaluator ***shall examine*** the security objectives rationale to determine that for each assumption for the operational environment it contains an appropriate justification that the security objectives for the operational environment are suitable to uphold that assumption.

406 If no security objectives for the operational environment trace back to the assumption, this work unit fails.

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

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

409 Note that the tracings from security objectives for the operational environment to assumptions provided in the security objectives rationale may be a part of a justification, but do not constitute a justification by themselves. Even in the case that a security objective of the operational environment is merely a restatement of an assumption, a justification is required, but this justification may be as minimal as “Security Objective X directly upholds Assumption Y”.

10.7 Extended components definition (ASE_ECD)

10.7.1 Evaluation of sub-activity (ASE_ECD.1)

10.7.1.1 Objectives

410 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 CC Part 2 or CC Part 3 components.

10.7.1.2 Input

411 The evaluation evidence for this sub-activity is:

a) the ST.

10.7.1.3 Action ASE_ECD.1.1E

ASE_ECD.1.1C ***The statement of security requirements shall identify all extended security requirements.***

ASE_ECD.1-1 The evaluator ***shall check*** that all security requirements in the statement of security requirements that are not identified as extended requirements are present in CC Part 2 or in CC Part 3.

ASE_ECD.1.2C ***The extended components definition shall define an extended component for each extended security requirement.***

ASE_ECD.1-2 The evaluator ***shall check*** that the extended components definition defines an extended component for each extended security requirement.

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

413 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.

ASE_ECD.1.3C ***The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.***

ASE_ECD.1-3 The evaluator ***shall examine*** the extended components definition to determine that it describes how each extended component fits into the existing CC components, families, and classes.

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

415 The evaluator determines that each extended component is either:

a) a member of an existing CC Part 2 or CC Part 3 family, or

b) a member of a new family defined in the ST.

416 If the extended component is a member of an existing CC Part 2 or CC Part 3 family, the evaluator determines that the extended components definition adequately describes why the extended component should be a member of that family and how it relates to other components of that family.

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

418 If the ST defines new families, the evaluator determines that each new family is either:

a) a member of an existing CC Part 2 or CC Part 3 class, or

b) a member of a new class defined in the ST.

419 If the family is a member of an existing CC Part 2 or CC Part 3 class, the evaluator determines that the extended components definition adequately describes why the family should be a member of that class and how it relates to other families in that class.

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

ASE_ECD.1-4 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended component identifies all applicable dependencies of that component.

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

422 The evaluator confirms that no applicable dependencies have been overlooked by the ST author.

ASE_ECD.1.4C ***The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.***

ASE_ECD.1-5 The evaluator ***shall examine*** the extended components definition to determine that each extended functional component uses the existing CC Part 2 components as a model for presentation.

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

424 The evaluator determines that the extended functional component is consistent with CC Part 2 Section 7.1.3, Component structure.

425 If the extended functional component uses operations, the evaluator determines that the extended functional component is consistent with CC Part 1 Annex C.4, Operations.

426 If the extended functional component is hierarchical to an existing functional component, the evaluator determines that the extended functional component is consistent with CC Part 2 Section 7.2.1, Component changes highlighting.

ASE_ECD.1-6 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new functional family uses the existing CC functional families as a model for presentation.

427 If the ST does not define new functional families, this work unit is not applicable and therefore considered to be satisfied.

428 The evaluator determines that all new functional families are defined consistent with CC Part 2 Section 7.1.2, Family structure.

ASE_ECD.1-7 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new functional class uses the existing CC functional classes as a model for presentation.

429 If the ST does not define new functional classes, this work unit is not applicable and therefore considered to be satisfied.

430 The evaluator determines that all new functional classes are defined consistent with CC Part 2 Section 7.1.1, Class structure.

ASE_ECD.1-8 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended assurance component uses the existing CC Part 3 components as a model for presentation.

431 If the ST does not contain extended SARs, this work unit is not applicable and therefore considered to be satisfied.

432 The evaluator determines that the extended assurance component definition is consistent with CC Part 3 Section 7.1.3, Assurance component structure.

433 If the extended assurance component uses operations, the evaluator determines that the extended assurance component is consistent with CC Part 1 Section Annex C.4, Operations.

434 If the extended assurance component is hierarchical to an existing assurance component, the evaluator determines that the extended assurance component is consistent with CC Part 3 Section 7.1.3, Assurance component structure.

ASE_ECD.1-9 The evaluator ***shall examine*** the extended components definition to determine that, for each defined extended assurance component, applicable methodology has been provided.

435 If the ST does not contain extended SARs, this work unit is not applicable and therefore considered to be satisfied.

436 The evaluator determines that, for each evaluator action element of each extended SAR, one or more work units are provided and that successfully

performing all work units for a given evaluator action element will demonstrate that the element has been achieved.

ASE_ECD.1-10 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new assurance family uses the existing CC assurance families as a model for presentation.

437 If the ST does not define new assurance families, this work unit is not applicable and therefore considered to be satisfied.

438 The evaluator determines that all new assurance families are defined consistent with CC Part 3 Section 7.1.2, Assurance family structure.

ASE_ECD.1-11 The evaluator ***shall examine*** the extended components definition to determine that each definition of a new assurance class uses the existing CC assurance classes as a model for presentation.

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

440 The evaluator determines that all new assurance classes are defined consistent with CC Part 3 Section 7.1.1, Assurance class structure.

ASE_ECD.1.5C ***The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.***

ASE_ECD.1-12 The evaluator ***shall examine*** the extended components definition to determine that each element in each extended component is measurable and states objective evaluation requirements, such that conformance or nonconformance can be demonstrated.

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

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

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

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

10.7.1.4 Action ASE_ECD.1.2E

ASE_ECD.1-13 The evaluator ***shall examine*** the extended components definition to determine that each extended component can not be clearly expressed using existing components.

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

446 The evaluator should take components from CC Part 2 and CC Part 3, other extended components that have been defined in the ST, combinations of these components, and possible operations on these components into account when making this determination.

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

10.8 Security requirements (ASE_REQ)

10.8.1 Evaluation of sub-activity (ASE_REQ.1)

10.8.1.1 Objectives

448 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and well-defined and whether they are internally consistent.

10.8.1.2 Input

449 The evaluation evidence for this sub-activity is:

a) the ST.

10.8.1.3 Action ASE_REQ.1.1E

ASE_REQ.1.1C ***The statement of security requirements shall describe the SFRs and the SARs.***

ASE_REQ.1-1 The evaluator ***shall check*** that the statement of security requirements describes the SFRs.

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

a) by reference to an individual component in CC Part 2;

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

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

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

e) by reproduction in the ST.

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

ASE_REQ.1-2 The evaluator ***shall check*** that the statement of security requirements describes the SARs.

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

a) by reference to an individual component in CC Part 3;

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

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

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

e) by reproduction in the ST.

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

ASE_REQ.1.2C ***All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.***

ASE_REQ.1-3 The evaluator ***shall examine*** the ST to determine that all subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

454 The evaluator determines that the ST defines all:

- (types of) subjects and objects that are used in the SFRs;
- (types of) security attributes of subjects, users, objects, information, sessions and/or resources, possible values that these attributes may take and any relations between these values (e.g. top_secret is "higher" than secret);
- (types of) operations that are used in the SFRs, including the effects of these operations;
- (types of) external entities in the SFRs;
- other terms that are introduced in the SFRs and/or SARs by completing operations, if these terms are not immediately clear, or are used outside their dictionary definition.

455 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no misunderstanding may occur due to the introduction of vague terms. This work unit should not be taken into extremes, by forcing the ST writer to define every single word. The general audience of a set of security requirements should be assumed to have a reasonable knowledge of IT, security and Common Criteria.

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

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

ASE_REQ.1.3C ***The statement of security requirements shall identify all operations on the security requirements.***

ASE_REQ.1-4 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

458 The evaluator determines that all operations are identified in each SFR or SAR where such an operation is used. Identification may be achieved by typographical distinctions, or by explicit identification in the surrounding text, or by any other distinctive means.

ASE_REQ.1.4C ***All operations shall be performed correctly.***

ASE_REQ.1-5 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

459 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.1-6 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration operations are performed correctly.

460 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.1-7 The evaluator ***shall examine*** the statement of security requirements to determine that all selection operations are performed correctly.

461 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.1-8 The evaluator ***shall examine*** the statement of security requirements to determine that all refinement operations are performed correctly.

462 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.1.5C ***Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.***

ASE_REQ.1-9 The evaluator ***shall examine*** the statement of security requirements to determine that each dependency of the security requirements is either satisfied, or that a security requirements rationale is provided which justifies the dependency not being satisfied.

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

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

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

b) that the dependency has been addressed by the operational environment of the TOE, in which case the justification should

describe how the security objectives for the operational environment address this dependency.

ASE_REQ.1.6C ***The statement of security requirements shall be internally consistent.***

ASE_REQ.1-10 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

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

467 Some possible conflicts are:

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

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

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

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

10.8.2 Evaluation of sub-activity (ASE_REQ.2)

10.8.2.1 Objectives

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

10.8.2.2 Input

469 The evaluation evidence for this sub-activity is:

a) the ST.

10.8.2.3 Action ASE_REQ.2.1E

ASE_REQ.2.1C ***The statement of security requirements shall describe the SFRs and the SARs.***

ASE_REQ.2-1 The evaluator ***shall check*** that the statement of security requirements describes the SFRs.

470 The evaluator determines that each SFRs is identified by one of the following means:

a) by reference to an individual component in CC Part 2;

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

c) by reference to an individual component in a PP that the ST claims to be conformant with;

d) by reference to an individual component in a security requirements package that the ST claims to be conformant with;

e) by reproduction in the ST.

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

ASE_REQ.2-2 The evaluator ***shall check*** that the statement of security requirements describes the SARs.

472 The evaluator determines that all SARs are identified by one of the following means:

a) by reference to an individual component in CC Part 3;

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

c) by reference to an individual component in a PP that the ST claims to be conformant with;

d) by reference to an individual component in a security requirements package that the ST claims to be conformant with;

e) by reproduction in the ST.

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

ASE_REQ.2.2C ***All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.***

ASE_REQ.2-3 The evaluator ***shall examine*** the ST to determine that all subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs are defined.

474 The evaluator determines that the ST defines all:

- (types of) subjects and objects that are used in the SFRs;
- (types of) security attributes of subjects, users, objects, information, sessions and/or resources, possible values that these attributes may take and any relations between these values (e.g. top_secret is "higher" than secret);
- (types of) operations that are used in the SFRs, including the effects of these operations;
- (types of) external entities in the SFRs;
- other terms that are introduced in the SFRs and/or SARs by completing operations, if these terms are not immediately clear, or are used outside their dictionary definition.

475 The goal of this work unit is to ensure that the SFRs and SARs are well-defined and that no misunderstanding may occur due to the introduction of vague terms. This work unit should not be taken into extremes, by forcing the ST writer to define every single word. The general audience of a set of security requirements should be assumed to have a reasonable knowledge of IT, security and Common Criteria.

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

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

ASE_REQ.2.3C ***The statement of security requirements shall identify all operations on the security requirements.***

ASE_REQ.2-4 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

478 The evaluator determines that all operations are identified in each SFR or SAR where such an operation is used. Identification may be achieved by typographical distinctions, or by explicit identification in the surrounding text, or by any other distinctive means.

ASE_REQ.2.4C ***All operations shall be performed correctly.***

ASE_REQ.2-5 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

479 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.2-6 The evaluator ***shall examine*** the statement of security requirements to determine that all iteration operations are performed correctly.

480 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.2-7 The evaluator ***shall examine*** the statement of security requirements to determine that all selection operations are performed correctly.

481 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.2-8 The evaluator ***shall examine*** the statement of security requirements to determine that all refinement operations are performed correctly.

482 Guidance on the correct performance of operations may be found in CC Part 1 Annex Annex C.4, Operations.

ASE_REQ.2.5C ***Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.***

ASE_REQ.2-9 The evaluator ***shall examine*** the statement of security requirements to determine that each dependency of the security requirements is either satisfied, or that the security requirements rationale justifies the dependency not being satisfied.

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

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

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

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

ASE_REQ.2.6C ***The security requirements rationale shall trace each SFR back to the security objectives for the TOE.***

ASE_REQ.2-10 The evaluator ***shall check*** that the security requirements rationale traces each SFR back to the security objectives for the TOE.

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

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

ASE_REQ.2.7C ***The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.***

ASE_REQ.2-11 The evaluator ***shall examine*** the security requirements rationale to determine that for each security objective for the TOE it demonstrates that the SFRs are suitable to meet that security objective for the TOE.

487 If no SFRs trace back to the security objective for the TOE, this work unit fails.

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

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

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

ASE_REQ.2.8C ***The security requirements rationale shall explain why the SARs were chosen.***

ASE_REQ.2-12 The evaluator ***shall check*** that the security requirements rationale explains why the SARs were chosen.

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

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

ASE_REQ.2.9C ***The statement of security requirements shall be internally consistent.***

ASE_REQ.2-13 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

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

495 Some possible conflicts are:

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

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

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

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

10.9 TOE summary specification (ASE_TSS)

10.9.1 Evaluation of sub-activity (ASE_TSS.1)

10.9.1.1 Objectives

496 The objective of this sub-activity is to determine whether the TOE summary specification addresses all SFRs, and whether the TOE summary specification is consistent with other narrative descriptions of the TOE.

10.9.1.2 Input

497 The evaluation evidence for this sub-activity is:

a) the ST.

10.9.1.3 Action ASE_TSS.1.1E

ASE_TSS.1.1C ***The TOE summary specification shall describe how the TOE meets each SFR.***

ASE_TSS.1-1 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE meets each SFR.

498 The evaluator determines that the TOE summary specification provides, for each SFR from the statement of security requirements, a description on how that SFR is met.

499 The evaluator is reminded that the objective of each description is to provide potential consumers of the TOE with a high-level view of how the developer intends to satisfy each SFR and that the descriptions therefore should not be overly detailed.

500 For a composed TOE, the evaluator also determines that it is clear which component provides each SFR or how the components combine to meet each SFR.

10.9.1.4 Action ASE_TSS.1.2E

ASE_TSS.1-2 The evaluator ***shall examine*** the TOE summary specification to determine that it is consistent with the TOE overview and the TOE description.

501 The TOE overview, TOE description, and TOE summary specification describe the TOE in a narrative form at increasing levels of detail. These descriptions therefore need to be consistent.

10.9.2 Evaluation of sub-activity (ASE_TSS.2)

10.9.2.1 Objectives

502 The objective of this sub-activity is to determine whether the TOE summary specification addresses all SFRs, whether the TOE summary specification

addresses interference, logical tampering and bypass, and whether the TOE summary specification is consistent with other narrative descriptions of the TOE.

10.9.2.2 Input

503 The evaluation evidence for this sub-activity is:

a) the ST.

10.9.2.3 Action ASE_TSS.2.1E

ASE_TSS.2.1C ***The TOE summary specification shall describe how the TOE meets each SFR.***

ASE_TSS.2-1 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE meets each SFR.

504 The evaluator determines that the TOE summary specification provides, for each SFR from the statement of security requirements, a description on how that SFR is met.

505 The evaluator is reminded that the objective of each description is to provide potential consumers of the TOE with a high-level view of how the developer intends to satisfy each SFR and that the descriptions therefore should not be overly detailed.

506 For a composed TOE, the evaluator also determines that it is clear which component provides each SFR or how the components combine to meet each SFR.

ASE_TSS.2.2C ***The TOE summary specification shall describe how the TOE protects itself against interference and logical tampering.***

ASE_TSS.2-2 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE protects itself against interference and logical tampering.

507 The evaluator is reminded that the objective of each description is to provide potential consumers of the TOE with a high-level view of how the developer intends to provide protection against interference and logical tampering and that the descriptions therefore should not be overly detailed.

508 For a composed TOE, the evaluator also determines that it is clear which component provides the protection or how the components combine to provide protection.

ASE_TSS.2.3C ***The TOE summary specification shall describe how the TOE protects itself against bypass.***

ASE_TSS.2-3 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE protects itself against bypass.

509 The evaluator is reminded that the objective of each description is to provide potential consumers of the TOE with a high-level view of how the developer intends to provide protection against bypass and that the descriptions therefore should not be overly detailed.

510 For a composed TOE, the evaluator also determines that it is clear which component provides the protection or how the components combine to provide protection.

10.9.2.4 Action ASE_TSS.2.2E

ASE_TSS.2-4 The evaluator ***shall examine*** the TOE summary specification to determine that it is consistent with the TOE overview and the TOE description.

511 The TOE overview, TOE description, and TOE summary specification describe the TOE in a narrative form at increasing levels of detail. These descriptions therefore need to be consistent.

11 Class ADV: Development

11.1 Introduction

512 The purpose of the development activity is to assess the design documentation in terms of its adequacy to understand how the TSF meets the SFRs and how the implementation of these SFRs cannot be tampered with or bypassed. This understanding is achieved through examination of increasingly refined descriptions of the TSF design documentation. Design documentation consists of a functional specification (which describes the interfaces of the TSF), a TOE design description (which describes the architecture of the TSF in terms of how it works in order to perform the functions related to the SFRs being claimed), and an implementation description (a source code level description). In addition, there is a security architecture description (which describes the architectural properties of the TSF to explain how its security enforcement cannot be compromised or bypassed), an internals description (which describes how the TSF was constructed in a manner that encourages understandability), and a security policy model (which formally describes the security policies enforced by the TSF).

11.2 Application notes

513 The CC requirements for design documentation are levelled by the amount, and detail of information provided, and the degree of formality of the presentation of the information. At lower levels, the most security-critical portions of the TSF are described with the most detail, while less security-critical portions of the TSF are merely summarised; added assurance is gained by increasing the amount of information about the most security-critical portions of the TSF, and increasing the details about the less security-critical portions. The most assurance is achieved when thorough details and information of all portions are provided.

514 The CC considers a document's degree of formality (that is, whether it is informal or semiformal) to be hierarchical. An informal document is one that is expressed in a natural language. The methodology does not dictate the specific language that must be used; that issue is left for the scheme. The following paragraphs differentiate the contents of the different informal documents.

515 A functional specification provides a description of the purpose and method-of-use of interfaces to the TSF. For example, if an operating system presents the user with a means of self-identification, of creating files, of modifying or deleting files, of setting permissions defining what other users may access files, and of communicating with remote machines, its functional specification would contain descriptions of each of these and how they are realised through interactions with the externally-visible interfaces to the TSF. If there is also audit functionality that detects and record the occurrences of such events, descriptions of this audit functionality would also be expected to

be part of the functional specification; while this functionality is technically not directly invoked by the user at the external interface, it certainly is affected by what occurs at the user's external interface.

516 A design description is expressed in terms of logical divisions (subsystems or components) that each provide a comprehensible service or function. For example, a firewall might be composed of subsystems that deal with packet filtering, with remote administration, with auditing, and with connection-level filtering. The design description of the firewall would describe the actions that are taken, in terms of what actions each subsystem takes when an incoming packet arrives at the firewall.

11.3 Security Architecture (ADV_ARC)

11.3.1 Evaluation of sub-activity (ADV_ARC.1)

11.3.1.1 Objectives

517 The objective of this sub-activity is to determine whether the TSF is structured such that it cannot be tampered with or bypassed, and whether TSFs that provide security domains isolate those domains from each other.

11.3.1.2 Input

518 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the implementation representation (if available);

f) the operational user guidance;

11.3.1.3 Application notes

519 The notions of self-protection, domain separation, and non-bypassability are distinct from security functionality expressed in Part 2 SFRs because self-protection and non-bypassability largely have no directly observable interface at the TSF. Rather, they are properties of the TSF that are achieved through the design of the TOE, and enforced by the correct implementation of that design. Also, the evaluation of these properties is less straight-forward than the evaluation of mechanisms; it is more difficult to check for the absence of functionality than for its presence. However, the determination that these properties are being satisfied is just as critical as the determination that the mechanisms are properly implemented.

520 The overall approach used is that the developer provides a TSF that meets the above-mentioned properties, and provides evidence (in the form of documentation) that can be analysed to show that the properties are indeed met. The evaluator has the responsibility for looking at the evidence and, coupled with other evidence delivered for the TOE, determining that the properties are achieved. The work units can be characterised as those detailing with what information has to be provided, and those dealing with the actual analysis the evaluator performs.

521 The security architecture description describes how domains are defined and how the TSF keeps them separate. It describes what prevents untrusted processes from getting to the TSF and modifying it. It describes what ensures that all resources under the TSF's control are adequately protected and that

all actions related to the SFRs are mediated by the TSF. It explains any role the environment plays in any of these (e.g. presuming it gets correctly invoked by its underlying environment, how is its security functionality invoked?). In short, it explains how the TOE is considered to be providing any kind of *security* service.

522 The analyses the evaluator performs must be done in the context of all of the development evidence provided for the TOE, at the level of detail the evidence is provided. At lower assurance levels there should not be the expectation that, for example, TSF self-protection is completely analysed, because only high-level design representations will be available. The evaluator also needs to be sure to use information gleaned from other portions of their analysis (e.g., analysis of the TOE design) in making their assessments for the properties being examined in the following work units.

11.3.1.4 Action ADV_ARC.1.1E

ADV_ARC.1.1C ***The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.***

ADV_ARC.1-1 The evaluator ***shall examine*** the security architecture description to determine that the information provided in the evidence is presented at a level of detail commensurate with the descriptions of the SFR-enforcing abstractions contained in the functional specification and TOE design document.

523 With respect to the functional specification, the evaluator should ensure that the self-protection functionality described cover those effects that are evident at the TSFI. Such a description might include protection placed upon the executable images of the TSF, and protection placed on objects (e.g., files used by the TSF). The evaluator ensures that the functionality that might be invoked through the TSFI is described.

524 If Evaluation of sub-activity (ADV_TDS.1) or Evaluation of sub-activity (ADV_TDS.2) is included, the evaluator ensures the security architecture description contains information on how any subsystems that contribute to TSF domain separation work.

525 If Evaluation of sub-activity (ADV_TDS.3) or higher is available, the evaluator ensures that the security architecture description also contains implementation-dependent information. For example, such a description might contain information pertaining to coding conventions for parameter checking that would prevent TSF compromises (e.g. buffer overflows), and information on stack management for call and return operations. The evaluator checks the descriptions of the mechanisms to ensure that the level of detail is such that there is little ambiguity between the description in the security architecture description and the implementation representation.

526 This work unit fails if the security architecture description mentions any module, subsystem, or interface that is not described in the functional specification or TOE design document.

ADV_ARC.1.2C ***The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.***

ADV_ARC.1-2 The evaluator ***shall examine*** the security architecture description to determine that it describes the security domains maintained by the TSF.

527 Security domains refer to environments supplied by the TSF for use by potentially-harmful entities; for example, a typical secure operating system supplies a set of resources (address space, per-process environment variables) for use by processes with limited access rights and security properties. The evaluator determines that the developer's description of the security domains takes into account all of the SFRs claimed by the TOE.

528 For some TOEs such domains do not exist because all of the interactions available to users are severely constrained by the TSF. A packet-filter firewall is an example of such a TOE. Users on the LAN or WAN do not interact with the TOE, so there need be no security domains; there are only data structures maintained by the TSF to keep the users' packets separated. The evaluator ensures that any claim that there are no domains is supported by the evidence and that no such domains are, in fact, available.

ADV_ARC.1.3C ***The security architecture description shall describe how the TSF initialisation process is secure.***

ADV_ARC.1-3 The evaluator ***shall examine*** the security architecture description to determine that the initialisation process preserves security.

529 The information provided in the security architecture description relating to TSF initialisation is directed at the TOE components that are involved in bringing the TSF into an initial secure state (i.e. when all parts of the TSF are operational) when power-on or a reset is applied. This discussion in the security architecture description should list the system initialisation components and the processing that occurs in transitioning from the “down” state to the initial secure state.

530 It is often the case that the components that perform this initialisation function are not accessible after the secure state is achieved; if this is the case then the architectural design identifies the components and explains how they are not reachable by untrusted entities after the TSF has been established. In this respect, the property that needs to be preserved is that these components either 1) cannot be accessed by untrusted entities after the secure state is achieved, or 2) if they provide interfaces to untrusted entities, these TSFI cannot be used to tamper with the TSF.

531 The TOE components related to TSF initialisation, then, are treated themselves as part of the TSF, and analysed from that perspective. It should be noted that even though these are treated as part of the TSF, it is likely that

a justification (as allowed by TSF internals (ADV_INT)) can be made that they do not have to meet the internal structuring requirements of ADV_INT.

ADV_ARC.1.4C ***The security architecture description shall demonstrate that the TSF protects itself from tampering.***

ADV_ARC.1-4 The evaluator ***shall examine*** the security architecture description to determine that it contains information sufficient to support a determination that the TSF is able to protect itself from tampering by untrusted active entities.

532 ”Self-protection” refers to the ability of the TSF to protect itself from manipulation from external entities that may result in changes to the TSF. For TOEs that have dependencies on other IT entities, it is often the case that the TOE uses services supplied by the other IT entities in order to perform its functions. In such cases, the TSF alone does not protect itself because it depends on the other IT entities to provide some of the protection. For the purposes of the security architecture description, the notion of *self-protection* applies only to the services provided by the TSF through its TSFI, and not to services provided by underlying IT entities that it uses.

533 Self-protection is typically achieved by a variety of means, ranging from physical and logical restrictions on access to the TOE; to hardware-based means (e.g. “execution rings” and memory management functionality); to software-based means (e.g. boundary checking of inputs on a trusted server). The evaluator determines that all such mechanisms are described.

534 The evaluator determines that the design description covers how user input is handled by the TSF in such a way that the TSF does not subject itself to being corrupted by that user input. For example, the TSF might implement the notion of privilege and protect itself by using privileged-mode routines to handle user data. The TSF might make use of processor-based separation mechanisms such as privilege levels or rings. The TSF might implement software protection constructs or coding conventions that contribute to implementing separation of software domains, perhaps by delineating user address space from system address space. And the TSF might have reliance its environment to provide some support to the protection of the TSF.

535 All of the mechanisms contributing to the domain separation functions are described. The evaluator should use knowledge gained from other evidence (functional specification, TOE design, TSF internals description, other parts of the security architecture description, or implementation representation, as included in the assurance package for the TOE) in determining if any functionality contributing to self-protection was described that is not present in the security architecture description.

536 Accuracy of the description of the self-protection mechanisms is the property that the description faithfully describes what is implemented. The evaluator should use other evidence (functional specification, TOE design, TSF Internals documentation, other parts of the security architectural description, implementation representation, as included in the ST for the TOE) in

determining whether there are discrepancies in any descriptions of the self-protection mechanisms. If Implementation representation (ADV_IMP) is included in the assurance package for the TOE, the evaluator will choose a sample of the implementation representation; the evaluator should also ensure that the descriptions are accurate for the sample chosen. If an evaluator cannot understand how a certain self-protection mechanism works or could work in the system architecture, it may be the case that the description is not accurate.

ADV_ARC.1.5C ***The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.***

ADV_ARC.1-5 The evaluator ***shall examine*** the security architecture description to determine that it presents an analysis that adequately describes how the SFR-enforcing mechanisms cannot be bypassed.

537 Non-bypassability is a property that the security functionality of the TSF (as specified by the SFRs) is always invoked. For example, if access control to files is specified as a capability of the TSF via an SFR, there must be no interfaces through which files can be accessed without invoking the TSF's access control mechanism (such as an interface through which a raw disk access takes place).

538 Describing how the TSF mechanisms cannot be bypassed generally requires a systematic argument based on the TSF and the TSFIs. The description of how the TSF works (contained in the design decomposition evidence, such as the functional specification, TOE design documentation) - along with the information in the TSS - provides the background necessary for the evaluator to understand what resources are being protected and what security functions are being provided. The functional specification provides descriptions of the TSFIs through which the resources/functions are accessed.

539 The evaluator assesses the description provided (and other information provided by the developer, such as the functional specification) to ensure that no available interface can be used to bypass the TSF. This means that every available interface must be either unrelated to the SFRs that are claimed in the ST (and does not interact with anything that is used to satisfy SFRs) or else uses the security functionality that is described in other development evidence in the manner described. For example, a game would likely be unrelated to the SFRs, so there must be an explanation of how it cannot affect security. Access to user data, however, is likely to be related to access control SFRs, so the explanation would describe how the security functionality works when invoked through the data-access interfaces. Such a description in needed for every available interface.

540 An example of a description follows. Suppose the TSF provides file protection. Further suppose that although the “traditional” system call TSFIs for open, read, and write invoke the file protection mechanism described in the TOE design, there exists a TSFI that allows access to a batch job facility (creating batch jobs, deleting jobs, modifying unprocessed jobs). The evaluator should be able to determine from the vendor-provided description

that this TSFI invokes the same protection mechanisms as do the "traditional" interfaces. This could be done, for example, by referencing the appropriate sections of the TOE design that discuss *how* the batch job facility TSFI achieves its security objectives.

541 Using this same example, suppose there is a TSFI whose sole purpose is to display the time of day. The evaluator should determine that the description adequately argues that this TSFI is not capable of manipulating any protected resources and should not invoke any security functionality.

542 Another example of bypass is when the TSF is supposed to maintain confidentiality of a cryptographic key (one is allowed to use it for cryptographic operations, but is not allowed to read/write it). If an attacker has direct physical access to the device, he might be able to examine side-channels such as the power usage of the device, the exact timing of the device, or even any electromagnetic emanations of the device and, from this, infer the key.

543 If such side-channels may be present, the demonstration should address the mechanisms that prevent these side-channels from occurring, such as random internal clocks, dual-line technology etc. Verification of these mechanisms would be verified by a combination of purely design-based arguments and testing.

544 For a final example using security functionality rather than a protected resource, consider an ST that contains FCO_NRO.2 Enforced proof of origin, which requires that the TSF provides evidence of origination for information types specified in the ST. Suppose that the "information types" included all information that is sent by the TOE via e-mail. In this case the evaluator should examine the description to ensure that all TSFI that can be invoked to send e-mail perform the "evidence of origination generation" function are detailed. The description might point to user guidance to show all places where e-mail can originate (e.g., e-mail program, notification from scripts/batch jobs) and then how each of these places invokes the evidence generation function.

545 The evaluator should also ensure that the description is comprehensive, in that each interface is analysed with respect to the entire set of claimed SFRs. This may require the evaluator to examine supporting information (functional specification, TOE design, other parts of the security architectural description, operational user guidance, and perhaps even the implementation representation, as provided for the TOE) to determine that the description has correctly capture all aspects of an interface. The evaluator should consider what SFRs each TSFI might affect (from the description of the TSFI and its implementation in the supporting documentation), and then examine the description to determine whether it covers those aspects.

11.4 Functional specification (ADV_FSP)

11.4.1 Evaluation of sub-activity (ADV_FSP.1)

11.4.1.1 Objectives

546 The objective of this sub-activity is to determine whether the developer has provided a high-level description of at least the SFR-enforcing and SFR-supporting TSFI, in terms of descriptions of their parameters. There is no other required evidence that can be expected to be available to measure the accuracy of these descriptions; the evaluator merely ensures the descriptions seem plausible.

11.4.1.2 Input

547 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the operational user guidance;

11.4.1.3 Action ADV_FSP.1.1E

ADV_FSP.1.1C ***The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.***

ADV_FSP.1-1 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each SFR-supporting and SFR-enforcing TSFI.

548 The purpose of a TSFI is a general statement summarising the functionality provided by the interface. It is not intended to be a complete statement of the actions and results related to the interface, but rather a statement to help the reader understand in general what the interface is intended to be used for. The evaluator should not only determine that the purpose exists, but also that it accurately reflects the TSFI by taking into account other information about the interface, such as the description of the parameters; this can be done in association with other work units for this component.

549 If an action available through an interface can be traced to one of the SFRs levied on the TSF), then that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but also refer to any functionality specified by one of the SFRs contained in the ST. Note that it is possible that an interface may have various actions and results, some of which may be SFR-enforcing and some of which may not.

550 Interfaces to (or actions available through an interface relating to) actions that SFR-enforcing functionality depends on, but need only to function correctly in order for the security policies of the TOE to be preserved, are termed *SFR supporting*. Interfaces to actions on which SFR-enforcing functionality has no dependence are termed *SFR non-interfering*.

551 It should be noted that in order for an interface to be SFR supporting or SFR non-interfering it must have *no* SFR-enforcing actions or results. In contrast, an SFR-enforcing interface may have SFR-supporting actions (for example, the ability to set the system clock may be an SFR-enforcing action of an interface, but if that same interface is used to display the system date that action may only be SFR supporting). An example of a purely SFR-supporting interface is a system call interface that is used both by untrusted users and by a portion of the TSF that is running in user mode.

552 At this level, it is unlikely that a developer will have expended effort to label interfaces as SFR-enforcing and SFR-supporting. In the case that this has been done, the evaluator should verify to the extent that supporting documentation (e.g., operational user guidance) allows that this identification is correct. Note that this identification activity is necessary for several work units for this component.

553 In the more likely case that the developer has not labelled the interfaces, the evaluator must perform their own identification of the interfaces first, and then determine whether the required information (for this work unit, the purpose) is present. Again, because of the lack of supporting evidence this identification will be difficult and have low assurance that all appropriate interfaces have been correctly identified, but nonetheless the evaluator examines other evidence available for the TOE to ensure as complete coverage as is possible.

ADV_FSP.1.2C ***The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.***

ADV_FSP.1-2 The evaluator ***shall examine*** the functional specification to determine that the method of use for each SFR-supporting and SFR-enforcing TSFI is given.

554 See work unit ADV_FSP.1-1 for a discussion on the identification of SFR-supporting and SFR-enforcing TSFI.

555 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the actions and obtain the results associated with the TSFI. The evaluator should be able to determine, from reading this material in the functional specification, how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each TSFI, as it may be possible to describe in general how kernel calls are invoked, for instance, and then identify each interface using that general style. Different types of interfaces will require different method of use specifications. APIs, network protocol interfaces, system configuration parameters, and hardware bus interfaces all have very different methods of use, and this should be taken into account by the developer when developing the functional specification, as well as by the evaluator evaluating the functional specification.

556 For administrative interfaces whose functionality is documented as being inaccessible to untrusted users, the evaluator ensures that the method of making the functions inaccessible is described in the functional specification.

It should be noted that this inaccessibility should be tested by the developer in their test suite.

ADV_FSP.1-3 The evaluator ***shall examine*** the presentation of the TSFI to determine that it identifies all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

557 See work unit ADV_FSP.1-1 for a discussion on the identification of SFR-supporting and SFR-enforcing TSFI.

558 The evaluator examines the functional specification to ensure that all of the parameters are described for identified TSFI. Parameters are explicit inputs or outputs to an interface that control the behaviour of that interface. For examples, parameters are the arguments supplied to an API; the various fields in packet for a given network protocol; the individual key values in the Windows Registry; the signals across a set of pins on a chip; etc.

559 While difficult to obtain much assurance that all parameters for the applicable TSFI have been identified, the evaluator should also check other evidence provided for the evaluation (e.g., operational user guidance) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.1.3C ***The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.***

ADV_FSP.1-4 The evaluator ***shall examine*** the rationale provided by the developer for the implicit categorisation of interfaces as SFR-non-interfering to determine that it is accurate.

560 In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, this work unit should be considered satisfied.

561 This work unit is intended to apply to cases where the developer has not described a portion of the TSFI, claiming that it is SFR-non-interfering and therefore not subject to other requirements of this component. In such a case, the developer provides a rationale for this characterisation in sufficient detail such that the evaluator understands the rationale, the characteristics of the interfaces affected (e.g., their high-level function with respect to the TOE, such as “colour palette manipulation”), and that the claim that these are SFR-non-interfering is supported. Given the level of assurance the evaluator should not expect more detail than is provided for the non-SFR-non-interfering interfaces, and in fact the detail should be much less. In most cases, individual interfaces should not need to be addressed in the developer-provided rationale section.

ADV_FSP.1.4C ***The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.***

ADV_FSP.1-5 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

562 The tracing is provided by the developer to serve as a guide to which SFRs are related to which TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the following work units, in which the evaluator verifies its completeness and accuracy.

11.4.1.4 Action ADV_FSP.1.2E

ADV_FSP.1-6 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

563 To ensure that all SFRs are covered by the functional specification, as well as the test coverage analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.1-5 a map between the TOE security functional requirements and the TSFI. Note that this map may have to be at a level of detail below the component or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

564 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of parameters for a given interface.

565 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions, and error messages associated with TSFIs must be fully specified, the evaluator should be able to determine if all aspects of an SFR appear to be implemented at the interface level.

ADV_FSP.1-7 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

566 For each functional requirement in the ST that results in effects visible at the TSF boundary, the information in the associated TSFI for that requirement specifies the required functionality described by the requirement. For example, if the ST contains a requirement for access control lists, and the only TSFI that map to that requirement specify functionality for Unix-style protection bits, then the functionality specification is not accurate with respect to the requirements.

567 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST.

11.4.2 Evaluation of sub-activity (ADV_FSP.2)

11.4.2.1 Objectives

568 The objective of this sub-activity is to determine whether the developer has provided a description of the TSFIs in terms of their purpose, method of use, and parameters. In addition, the SFR-enforcing actions, results and error messages of each TSFI that is SFR-enforcing are also described.

11.4.2.2 Input

569 The evaluation evidence for this sub-activity that is required by the work-units is:

a) the ST;

b) the functional specification;

c) the TOE design.

570 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

a) the security architecture description;

b) the operational user guidance;

11.4.2.3 Action ADV_FSP.2.1E

ADV_FSP.2.1C ***The functional specification shall completely represent the TSF*.**

ADV_FSP.2-1 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

571 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity. The TSF must be identified (done as part of the TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be done at a high level to ensure that no large groups of interfaces have been missed (network protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the functional specification proceeds.

572 In making an assessment for this work unit, the evaluator determines that all portions of the TSF are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF should have a corresponding

interface description, or if there are no corresponding interfaces for a portion of the TSF, the evaluator determines that that is acceptable.

ADV_FSP.2.2C ***The functional specification shall describe the purpose and method of use for all TSFI.***

ADV_FSP.2-2 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each TSFI.

573 The purpose of a TSFI is a general statement summarising the functionality provided by the interface. It is not intended to be a complete statement of the actions and results related to the interface, but rather a statement to help the reader understand in general what the interface is intended to be used for. The evaluator should not only determine that the purpose exists, but also that it accurately reflects the TSFI by taking into account other information about the interface, such as the description of actions and error messages.

ADV_FSP.2-3 The evaluator ***shall examine*** the functional specification to determine that the method of use for each TSFI is given.

574 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the actions and obtain the results associated with the TSFI. The evaluator should be able to determine, from reading this material in the functional specification, how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each TSFI, as it may be possible to describe in general how kernel calls are invoked, for instance, and then identify each interface using that general style. Different types of interfaces will require different method of use specifications. APIs, network protocol interfaces, system configuration parameters, and hardware bus interfaces all have very different methods of use, and this should be taken into account by the developer when developing the functional specification, as well as by the evaluator evaluating the functional specification.

575 For administrative interfaces whose functionality is documented as being inaccessible to untrusted users, the evaluator ensures that the method of making the functions inaccessible is described in the functional specification. It should be noted that this inaccessibility should be tested by the developer in their test suite.

576 The evaluator should not only determine that the set of method of use descriptions exist, but also that they accurately cover each TSFI.

ADV_FSP.2.3C ***The functional specification shall identify and describe all parameters associated with each TSFI.***

ADV_FSP.2-4 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies all parameters associated with every TSFI.

577 The evaluator examines the functional specification to ensure that all of the parameters are described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the behaviour of that interface. For

examples, parameters are the arguments supplied to an API; the various fields in packet for a given network protocol; the individual key values in the Windows Registry; the signals across a set of pins on a chip; etc.

578 In order to determine that all of the parameters are present in the TSFI, the evaluator should examine the rest of the interface description (actions, error messages, etc.) to determine if the effects of the parameter are accounted for in the description. The evaluator should also check other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.2-5 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all parameters associated with every TSFI.

579 Once all of the parameters have been identified, the evaluator needs to ensure that they are accurately described, and that the description of the parameters is complete. A parameter description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)* could be described as having “parameter i which is an integer"; this is not an acceptable parameter description. A description such as “parameter i is an integer that indicates the number of users currently logged in to the system” is much more acceptable.

580 In order to determine that the description of the parameters is complete, the evaluator should examine the rest of the interface description (purpose, method of use, actions, error messages, etc.) to determine if the descriptions of the parameter(s) are accounted for in the description. The evaluator should also check other evidence provided (e.g., TOE design, architectural design, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.2.4C ***For SFR-enforcing TSFIs, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.***

ADV_FSP.2-6 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes the SFR-enforcing actions associated with the SFR-enforcing TSFIs.

581 If an action available through an interface plays a role in enforcing any security policy on the TOE (that is, if one of the actions of the interface can be traced to one of the SFRs levied on the TSF), then that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but also refer to any functionality specified by one of the SFRs contained in the ST. Note that it is possible that an interface may have various actions and results, some of which may be SFR-enforcing and some of which may not.

582 The developer is not required to “label” interfaces as SFR-enforcing, and likewise is not required to identify actions available through an interface as

SFR-enforcing. It is the evaluator's responsibility to examine the evidence provided by the developer and determine that the required information is present. In the case where the developer has identified the SFR-enforcing TSFI and SFR-enforcing actions available through those TSFI, the evaluator must judge completeness and accuracy based on other information supplied for the evaluation (e.g., TOE design, security architecture description, operational user guidance), and on the other information presented for the interfaces (parameters and parameter descriptions, error messages, etc.).

583 In this case (where the developer has provided only the SFR-enforcing information for SFR-enforcing TSFI) the evaluator also ensures that no interfaces have been mis-categorised. This is done by examining other information supplied for the evaluation (e.g., TOE design, security architecture description, operational user guidance), and the other information presented for the interfaces (parameters and parameter descriptions, for example) not labelled as SFR-enforcing.

584 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.

585 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).

586 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.

ADV_FSP.2.5C ***For SFR-enforcing TSFIs, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.***

ADV_FSP.2-7 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes error messages that may result from SFR-enforcing actions associated with each SFR-enforcing TSFI.

587 This work unit should be performed in conjunction with, or after, work unit ADV_FSP.2-6 in order to ensure the set of SFR-enforcing TSFI and SFR-enforcing actions is correctly identified. The developer may provide more information than is required (for example, all error messages associated with

each interface), in which the case the evaluator should restrict their assessment of completeness and accuracy to only those that they determine to be associated with SFR-enforcing actions of SFR-enforcing TSFI.

588 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.

589 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), TOE design (ADV_TDS)) related that TSFI to determine if the description is accurate.

590 In order to determine that the description of the error messages of a TSFI is accurate and complete, the evaluator measures the interface description against the other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance), as well as other evidence available for that TSFI (parameters, analysis from work unit ADV_FSP.2-6).

ADV_FSP.2.6C ***The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.***

ADV_FSP.2-8 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

591 The tracing is provided by the developer to serve as a guide to which SFRs are related to which TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the following work units, in which the evaluator verifies its completeness and accuracy.

11.4.2.4 Action ADV_FSP.2.2E

ADV_FSP.2-9 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

592 To ensure that all SFRs are covered by the functional specification, as well as the test coverage analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.2-8 a map between the TOE security functional requirements and the TSFI. Note that this map may have to be at a level of detail below the component or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

593 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of parameters for a given interface.

594 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions, and error messages associated with TSFIs must be fully specified, the evaluator should be able to determine if all aspects of an SFR appear to be implemented at the interface level.

ADV_FSP.2-10 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

595 For each functional requirement in the ST that results in effects visible at the TSF boundary, the information in the associated TSFI for that requirement specifies the required functionality described by the requirement. For example, if the ST contains a requirement for access control lists, and the only TSFI that map to that requirement specify functionality for Unix-style protection bits, then the functionality specification is not accurate with respect to the requirements.

596 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST.

11.4.3 Evaluation of sub-activity (ADV_FSP.3)

11.4.3.1 Objectives

597 The objective of this sub-activity is to determine whether the developer has provided a description of the TSFIs in terms of their purpose, method of use, and parameters. In addition, the actions, results and error messages of each

TSFI are also described sufficiently that it can be determined whether they are SFR-enforcing, with the SFR-enforcing TSFI being described in more detail than other TSFIs.

11.4.3.2 Input

598 The evaluation evidence for this sub-activity that is required by the work-units is:

a) the ST;

b) the functional specification;

c) the TOE design.

599 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

a) the security architecture description;

b) the implementation representation;

c) the TSF internals description;

d) the operational user guidance;

11.4.3.3 Action ADV_FSP.3.1E

ADV_FSP.3.1C ***The functional specification shall completely represent the TSF.***

ADV_FSP.3-1 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

600 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity. The TSF must be identified (done as part of the TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be done at a high level to ensure that no large groups of interfaces have been missed (network protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the functional specification proceeds.

601 In making an assessment for this work unit, the evaluator determines that all portions of the TSF are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF should have a corresponding interface description, or if there are no corresponding interfaces for a portion of the TSF, the evaluator determines that that is acceptable.

ADV_FSP.3.2C ***The functional specification shall describe the purpose and method of use for all TSFI.***

ADV_FSP.3-2 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each TSFI.

602 The purpose of a TSFI is a general statement summarising the functionality provided by the interface. It is not intended to be a complete statement of the actions and results related to the interface, but rather a statement to help the reader understand in general what the interface is intended to be used for. The evaluator should not only determine that the purpose exists, but also that it accurately reflects the TSFI by taking into account other information about the interface, such as the description of actions and error messages.

ADV_FSP.3-3 The evaluator ***shall examine*** the functional specification to determine that the method of use for each TSFI is given.

603 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the actions and obtain the results associated with the TSFI. The evaluator should be able to determine, from reading this material in the functional specification, how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each TSFI, as it may be possible to describe in general how kernel calls are invoked, for instance, and then identify each interface using that general style. Different types of interfaces will require different method of use specifications. APIs, network protocol interfaces, system configuration parameters, and hardware bus interfaces all have very different methods of use, and this should be taken into account by the developer when developing the functional specification, as well as by the evaluator evaluating the functional specification.

604 For administrative interfaces whose functionality is documented as being inaccessible to untrusted users, the evaluator ensures that the method of making the functions inaccessible is described in the functional specification. It should be noted that this inaccessibility should be tested by the developer in their test suite.

605 The evaluator should not only determine that the set of method of use descriptions exist, but also that they accurately cover each TSFI.

ADV_FSP.3.3C ***The functional specification shall identify and describe all parameters associated with each TSFI.***

ADV_FSP.3-4 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies all parameters associated with every TSFI.

606 The evaluator examines the functional specification to ensure that all of the parameters are described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the behaviour of that interface. For examples, parameters are the arguments supplied to an API; the various fields in packet for a given network protocol; the individual key values in the Windows Registry; the signals across a set of pins on a chip; etc.

607 In order to determine that all of the parameters are present in the TSFI, the evaluator should examine the rest of the interface description (actions, error messages, etc.) to determine if the effects of the parameter are accounted for in the description. The evaluator should also check other evidence provided for the evaluation (e.g., TOE design, security architecture description,

operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.3-5 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all parameters associated with every TSFI.

608 Once all of the parameters have been identified, the evaluator needs to ensure that they are accurately described, and that the description of the parameters is complete. A parameter description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)* could be described as having "parameter i which is an integer"; this is not an acceptable parameter description. A description such as "parameter i is an integer that indicates the number of users currently logged in to the system" is much more acceptable.

609 In order to determine that the description of the parameters is complete, the evaluator should examine the rest of the interface description (purpose, method of use, actions, error messages, etc.) to determine if the descriptions of the parameter(s) are accounted for in the description. The evaluator should also check other evidence provided (e.g., TOE design, architectural design, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.3.4C ***For SFR-enforcing TSFIs, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.***

ADV_FSP.3-6 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes the SFR-enforcing actions associated with the SFR-enforcing TSFIs.

610 If an action available through an interface plays a role in enforcing any security policy on the TOE (that is, if one of the actions of the interface can be traced to one of the SFRs levied on the TSF), then that interface is *SFR-enforcing*. Such policies are not limited to the access control policies, but also refer to any functionality specified by one of the SFRs contained in the ST. Note that it is possible that an interface may have various actions and results, some of which may be SFR-enforcing and some of which may not.

611 The developer is not required to "label" interfaces as SFR-enforcing, and likewise is not required to identify actions available through an interface as SFR-enforcing. It is the evaluator's responsibility to examine the evidence provided by the developer and determine that the required information is present. In the case where the developer has identified the SFR-enforcing TSFI and SFR-enforcing actions available through those TSFI, the evaluator must judge completeness and accuracy based on other information supplied for the evaluation (e.g., TOE design, security architecture description, operational user guidance), and on the other information presented for the interfaces (parameters and parameter descriptions, error messages, etc.).

612 In this case (developer has provided only the SFR-enforcing information for SFR-enforcing TSFI) the evaluator also ensures that no interfaces have been mis-categorised. This is done by examining other information supplied for the evaluation (e.g., TOE design, security architecture description, operational user guidance), and the other information presented for the interfaces (parameters and parameter descriptions, for example) not labelled as SFR-enforcing. The analysis done for work units ADV_FSP.3-7 and ADV_FSP.3-8 are also used in making this determination.

613 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.

614 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).

615 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.

ADV_FSP.3.5C ***For SFR-enforcing TSFIs, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions associated with invocation of the TSFI.***

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.

616 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 error messages associated with an SFR-enforcing TSFI must be described, not just those 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.

617 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.

618 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), TOE design (ADV_TDS)) related that TSFI to determine if the description is accurate.

619 In order to determine that the description of the error messages of a TSFI is accurate and complete, the evaluator measures the interface description against the other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance), as well as for other evidence supplied for that TSFI (description of SFR-enforcing actions, summary of non-SFR-enforcing actions and results).

ADV_FSP.3.6C ***The functional specification shall summarise the non-SFR-enforcing actions associated with each TSFI.***

ADV_FSP.3-8 The evaluator ***shall examine*** the presentation of the TSFI to determine that it summarises the non-SFR-enforcing actions associated with each TSFI.

620 The purpose of this work unit is to supplement the details about the SFR-enforcing actions (provided in work unit ADV_FSP.3-6) with a summary of the remaining actions (i.e., those that are not SFR-enforcing). This covers *all* non-SFR-enforcing actions, whether invokable through SFR-enforcing TSFI or through non-SFR-enforcing TSFI. Such a summary about all non-SFR-enforcing actions helps to provide a more complete picture of the functions provided by the TSF, and is to be used by the evaluator in determining whether an action or TSFI may have been mis-categorised.

621 The information to be provided is more abstract than that required for SFR-enforcing actions. While it should still be detailed enough so that the reader can understand what the action does, the description does not have to be detailed enough to support writing tests against it, for instance. For the evaluator, the key is that the information must be sufficient to make a positive determination that the action is non-SFR-enforcing. If that level of information is missing, the summary is insufficient and more information must be obtained.

ADV_FSP.3.7C ***The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.***

ADV_FSP.3-9 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

622 The tracing is provided by the developer to serve as a guide to which SFRs are related to which TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the following work units, in which the evaluator verifies its completeness and accuracy.

11.4.3.4 Action ADV_FSP.3.2E

ADV_FSP.3-10 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

623 To ensure that all SFRs are covered by the functional specification, as well as the test coverage analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.3-9 a map between the TOE security functional requirements and the TSFI. Note that this map may have to be at a level of detail below the component or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

624 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of parameters for a given interface.

625 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions, and error messages associated with TSFIs must be fully specified, the evaluator should be able to determine if all aspects of an SFR appear to be implemented at the interface level.

ADV_FSP.3-11 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

626 For each functional requirement in the ST that results in effects visible at the TSF boundary, the information in the associated TSFI for that requirement specifies the required functionality described by the requirement. For example, if the ST contains a requirement for access control lists, and the only TSFI that map to that requirement specify functionality for Unix-style protection bits, then the functionality specification is not accurate with respect to the requirements.

627 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST.

11.4.4 Evaluation of sub-activity (ADV_FSP.4)

11.4.4.1 Objectives

628 The objective of this sub-activity is to determine whether the developer has completely described all of the TSFI in a manner such that the evaluator is able to determine whether the TSFI are completely and accurately described, and appears to implement the security functional requirements of the ST.

11.4.4.2 Input

629 The evaluation evidence for this sub-activity that is required by the work-units is:

a) the ST;

b) the functional specification;

c) the TOE design.

630 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

a) the security architecture description;

b) the implementation representation;

c) the TSF internals description;

d) the operational user guidance;

11.4.4.3 Application notes

631 The functional specification describes the interfaces to the TSF (the TSFI) in a structured manner. Because of the dependency on Evaluation of sub-

activity (ADV_TDS.1), the evaluator is expected to have identified the TSF prior to beginning work on this sub-activity. Without firm knowledge of what comprises the TSF, it is not possible to assess the completeness of the TSFI.

632 In performing the various work units included in this family, the evaluator is asked to make assessments of accuracy and completeness of several factors (the TSFI itself, as well as the individual components (parameters, actions, error messages, etc.) of the TSFI). In doing this analysis, the evaluator is expected to use the documentation provided for the evaluation. This includes the ST, the TOE design, and may include other documentation such as the user and administrative guidance, security architecture description, and implementation representation. The documentation should be examined in an iterative fashion. The evaluator may read, for example, in the TOE design how a certain function is implemented, but see no way to invoke that function from the interface. This might cause the evaluator to question the completeness of a particular TSFI description, or whether an interface has been left out of the functional specification altogether. Describing analysis activities of this sort in the ETR is a key method in providing rationale that the work units have been performed appropriately.

633 It should be recognised that there exist functional requirements whose functionality is manifested wholly or in part architecturally, rather than through a specific mechanism. An example of this is the implementation of mechanisms implementing the Residual information protection (FDP_RIP) requirements. Such mechanisms typically are implemented to ensure a behaviour isn't present, which is difficult to test and typically is verified through analysis. In the cases where such functional requirements are included in the ST, it is expected that the evaluator recognise that there may be SFRs of this type that have no interfaces, and that this should not be considered a deficiency in the functional specification.

11.4.4.4 Action ADV_FSP.4.1E

ADV_FSP.4.1C ***The functional specification shall completely represent the TSF.***

ADV_FSP.4-1 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

634 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity. The TSF must be identified (done as part of the TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be done at a high level to ensure that no large groups of interfaces have been missed (network protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the functional specification proceeds.

635 In making an assessment for this work unit, the evaluator determines that all portions of the TSF are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF should have a corresponding

interface description, or if there are no corresponding interfaces for a portion of the TSF, the evaluator determines that that is acceptable.

ADV_FSP.4.2C ***The functional specification shall describe the purpose and method of use for all TSFI.***

ADV_FSP.4-2 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each TSFI.

636 The purpose of a TSFI is a general statement summarising the functionality provided by the interface. It is not intended to be a complete statement of the actions and results related to the interface, but rather a statement to help the reader understand in general what the interface is intended to be used for. The evaluator should not only determine that the purpose exists, but also that it accurately reflects the TSFI by taking into account other information about the interface, such as the description of actions and error messages.

ADV_FSP.4-3 The evaluator ***shall examine*** the functional specification to determine that the method of use for each TSFI is given.

637 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the actions and obtain the results associated with the TSFI. The evaluator should be able to determine, from reading this material in the functional specification, how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each TSFI, as it may be possible to describe in general how kernel calls are invoked, for instance, and then identify each interface using that general style. Different types of interfaces will require different method of use specifications. APIs, network protocol interfaces, system configuration parameters, and hardware bus interfaces all have very different methods of use, and this should be taken into account by the developer when developing the functional specification, as well as by the evaluator evaluating the functional specification.

638 For administrative interfaces whose functionality is documented as being inaccessible to untrusted users, the evaluator ensures that the method of making the functions inaccessible is described in the functional specification. It should be noted that this inaccessibility should be tested by the developer in their test suite.

639 The evaluator should not only determine that the set of method of use descriptions exist, but also that they accurately cover each TSFI.

ADV_FSP.4.3C ***The functional specification shall identify and describe all parameters associated with each TSFI.***

ADV_FSP.4-4 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies all parameters associated with every TSFI.

640 The evaluator examines the functional specification to ensure that all of the parameters are described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the behaviour of that interface. For

examples, parameters are the arguments supplied to an API; the various fields in packet for a given network protocol; the individual key values in the Windows Registry; the signals across a set of pins on a chip; etc.

641 In order to determine that all of the parameters are present in the TSFI, the evaluator should examine the rest of the interface description (actions, error messages, etc.) to determine if the effects of the parameter are accounted for in the description. The evaluator should also check other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.4-5 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all parameters associated with every TSFI.

642 Once all of the parameters have been identified, the evaluator needs to ensure that they are accurately described, and that the description of the parameters is complete. A parameter description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)* could be described as having "parameter i which is an integer"; this is not an acceptable parameter description. A description such as "parameter i is an integer that indicates the number of users currently logged in to the system" is much more acceptable.

643 In order to determine that the description of the parameters is complete, the evaluator should examine the rest of the interface description (purpose, method of use, actions, error messages, etc.) to determine if the descriptions of the parameter(s) are accounted for in the description. The evaluator should also check other evidence provided (e.g., TOE design, architectural design, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.4.4C ***The functional specification shall describe all actions associated with each TSFI.***

ADV_FSP.4-6 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all actions associated with every TSFI.

644 The evaluator checks to ensure that all of the actions are described. actions available through an interface describe what the interface does (as opposed to the TOE design, which describes how the actions are provided by the TSF).

645 Actions of an interface describe functionality that can be invoked through the interface, and can be categorised as *regular* actions, and *SFR-related* actions. Regular actions are descriptions of what the interface does. The amount of information provided for this description is dependant on the complexity of the interface. The SFR-related actions are those that are visible at any external interface (for instance, audit activity caused by the invocation of an interface (assuming audit requirements are included in the ST) should be

described, even though the result of that action is generally not visible through the invoked interface). Depending on the parameters of an interface, there may be many different actions able to be invoked through the interface (for instance, an API might have the first parameter be a “subcommand”, and the following parameters be specific to that subcommand. The IOCTL API in some Unix systems is an example of such an interface).

646 In order to determine that the description of the actions of a TSFI is complete, the evaluator should review the rest of the interface description (parameter descriptions, error messages, etc.) to determine if the actions described are accounted for. The evaluator should also analyse other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if there is evidence of actions that are described there but not in the functional specification.

ADV_FSP.4.5C ***The functional specification shall describe all direct error messages that may result from security enforcing effects and exceptions associated with an invocation of each TSFI.***

ADV_FSP.4-7 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all errors messages resulting from an invocation of each TSFI.

647 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.

648 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), TOE design (ADV_TDS)) related that TSFI to determine if the description is complete and accurate.

649 The evaluator determines that, for each TSFI, the exact set of error messages that can be returned on invoking that interface can be determined. The evaluator reviews the evidence provided for the interface to determine if the

set of errors seems complete. They cross-check this information with other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to ensure that there are no errors steaming from processing mentioned that are not included in the functional specification.

ADV_FSP.4-8 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes the meaning of all errors associated with each TSFI.

650 In order to determine accuracy, the evaluator must be able to understand meaning of the error. For example, if an interface returns a numeric code of 0, 1, or 2, the evaluator would not be able to understand the error if the functional specification only listed: “possible errors resulting from invocation of the *foo()* interface are 0, 1, or 2”. Instead the evaluator checks to ensure that the errors are described such as: “possible errors resulting from invocation of the *foo()* interface are 0 (processing successful), 1 (file not found), or 2 (incorrect filename specification)”.

651 In order to determine that the description of the errors due to invoking a TSFI is complete, the evaluator examines the rest of the interface description (parameter descriptions, actions, etc.) to determine if potential error conditions that might be caused by using such an interface are accounted for. The evaluator also checks other evidence provided for the evaluation (e.g. TOE design, security architecture description, operational user guidance, implementation representation) to see if error processing related to the TSFI is described there but is not described in the functional specification.

ADV_FSP.4.6C ***The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.***

ADV_FSP.4-9 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

652 The tracing is provided by the developer to serve as a guide to which SFRs are related to which TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the following work units, in which the evaluator verifies its completeness and accuracy.

11.4.4.5 Action ADV_FSP.4.2E

ADV_FSP.4-10 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

653 To ensure that all SFRs are covered by the functional specification, as well as the test coverage analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.4-9 a map between the TOE security functional requirements and the TSFI. Note that this map may have to be at a level of detail below the component or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

654 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of parameters for a given interface.

655 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions, and error messages associated with TSFIs must be fully specified, the evaluator should be able to determine if all aspects of an SFR appear to be implemented at the interface level.

ADV_FSP.4-11 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

656 For each functional requirement in the ST that results in effects visible at the TSF boundary, the information in the associated TSFI for that requirement specifies the required functionality described by the requirement. For example, if the ST contains a requirement for access control lists, and the only TSFI that map to that requirement specify functionality for Unix-style protection bits, then the functionality specification is not accurate with respect to the requirements.

657 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST.

11.4.5 Evaluation of sub-activity (ADV_FSP.5)

11.4.5.1 Objectives

658 The objective of this sub-activity is to determine whether the developer has completely described all of the TSFI in a manner such that the evaluator is able to determine whether the TSFI are completely and accurately described, and appears to implement the security functional requirements of the ST. The completeness of the interfaces is judged based upon the implementation representation.

11.4.5.2 Input

659 The evaluation evidence for this sub-activity that is required by the work-units is:

a) the ST;

b) the functional specification;

c) the TOE design;

d) the implementation representation.

660 The evaluation evidence for this sub-activity that is used if included in the ST for the TOE is:

a) the security architecture description;

b) the TSF internals description;

c) the formal security policy model;

d) the operational user guidance;

11.4.5.3 Action ADV_FSP.5.1E

ADV_FSP.5.1C ***The functional specification shall completely represent the TSF.***

ADV_FSP.5-1 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

661 The identification of the TSFI is a necessary prerequisite to all other activities in this sub-activity. The TSF must be identified (done as part of the TOE design (ADV_TDS) work units) in order to identify the TSFI. This activity can be done at a high level to ensure that no large groups of interfaces have been missed (network protocols, hardware interfaces, configuration files), or at a low level as the evaluation of the functional specification proceeds.

662 In making an assessment for this work unit, the evaluator determines that all portions of the TSF are addressed in terms of the interfaces listed in the functional specification. All portions of the TSF should have a corresponding interface description, or if there are no corresponding interfaces for a portion of the TSF, the evaluator determines that that is acceptable.

ADV_FSP.5.2C ***The functional specification shall describe the TSFI using a semi-formal style.***

ADV_FSP.5-2 The evaluator ***shall examine*** the functional specification to determine that it is presented using a semiformal style.

663 A semi-formal presentation is characterised by a standardised format with a well-defined syntax that reduces ambiguity that may occur in informal presentations. Since the intent of the semi-formal format is to enhance the reader's ability to understand the presentation, use of certain structured presentation methods (pseudo-code, flow charts, block diagrams) are appropriate, though not required.

664 For the purposes of this activity, the evaluator should ensure that the interface descriptions are formatted in a structured, consistent manner and use common terminology. A semiformal presentation of the interfaces also implies that the level of detail of the presentation for the interfaces is largely consistent across all TSFI. For the functional specification, it is acceptable to refer to external specifications for portions of the interface as long as those external specifications are themselves semiformal.

ADV_FSP.5.3C ***The functional specification shall describe the purpose and method of use for all TSFI.***

ADV_FSP.5-3 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each TSFI.

665 The purpose of a TSFI is a general statement summarising the functionality provided by the interface. It is not intended to be a complete statement of the actions and results related to the interface, but rather a statement to help the reader understand in general what the interface is intended to be used for. The evaluator should not only determine that the purpose exists, but also that it accurately reflects the TSFI by taking into account other information about the interface, such as the description of actions and error messages.

ADV_FSP.5-4 The evaluator ***shall examine*** the functional specification to determine that the method of use for each TSFI is given.

666 The method of use for a TSFI summarises how the interface is manipulated in order to invoke the actions and obtain the results associated with the TSFI. The evaluator should be able to determine, from reading this material in the functional specification, how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each TSFI, as it may be possible to describe in general how kernel calls are invoked, for instance, and then identify each interface using that general style. Different types of interfaces will require different method of use specifications. APIs, network protocol interfaces, system configuration parameters, and hardware bus interfaces all have very different methods of use, and this should be taken into account by the developer when developing the functional specification, as well as by the evaluator evaluating the functional specification.

667 For administrative interfaces whose functionality is documented as being inaccessible to untrusted users, the evaluator ensures that the method of making the functions inaccessible is described in the functional specification. It should be noted that this inaccessibility should be tested by the developer in their test suite.

668 The evaluator should not only determine that the set of method of use descriptions exist, but also that they accurately cover each TSFI.

ADV_FSP.5.4C ***The functional specification shall identify and describe all parameters associated with each TSFI.***

ADV_FSP.5-5 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely identifies all parameters associated with every TSFI.

669 The evaluator examines the functional specification to ensure that all of the parameters are described for each TSFI. Parameters are explicit inputs or outputs to an interface that control the behaviour of that interface. For examples, parameters are the arguments supplied to an API; the various fields in packet for a given network protocol; the individual key values in the Windows Registry; the signals across a set of pins on a chip; etc.

670 In order to determine that all of the parameters are present in the TSFI, the evaluator should examine the rest of the interface description (actions, error messages, etc.) to determine if the effects of the parameter are accounted for in the description. The evaluator should also check other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.5.5C ***The functional specification shall describe all actions associated with each TSFI.***

ADV_FSP.5-6 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all parameters associated with every TSFI.

671 Once all of the parameters have been identified, the evaluator needs to ensure that they are accurately described, and that the description of the parameters is complete. A parameter description tells what the parameter is in some meaningful way. For instance, the interface *foo(i)* could be described as having “parameter i which is an integer”; this is not an acceptable parameter description. A description such as “parameter i is an integer that indicates the number of users currently logged in to the system”. is much more acceptable.

672 In order to determine that the description of the parameters is complete, the evaluator should examine the rest of the interface description (purpose, method of use, actions, error messages, etc.) to determine if the descriptions of the parameter(s) are accounted for in the description. The evaluator should also check other evidence provided (e.g., TOE design, architectural design, operational user guidance, implementation representation) to see if behaviour or additional parameters are described there but not in the functional specification.

ADV_FSP.5-7 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all actions associated with every TSFI.

673 The evaluator checks to ensure that all of the actions are described. actions available through an interface describe what the interface does (as opposed to the TOE design, which describes how the actions are provided by the TSF).

674 actions of an interface describe functionality that can be invoked through the interface, and can be categorised as *regular* actions, and *SFR-related* actions. Regular actions are descriptions of what the interface does. The amount of information provided for this description is dependant on the complexity of the interface. The SFR-related actions are those that are visible at any external interface (for instance, audit activity caused by the invocation of an interface (assuming audit requirements are included in the ST) should be described, even though the result of that action is generally not visible through the invoked interface). Depending on the parameters of an interface, there may be many different actions able to be invoked through the interface (for instance, an API might have the first parameter be a "subcommand", and the following parameters be specific to that subcommand. The IOCTL API in some Unix systems is an example of such an interface).

675 In order to determine that the description of the actions of a TSFI is complete, the evaluator should review the rest of the interface description (parameter descriptions, error messages, etc.) to determine if the actions described are accounted for. The evaluator should also analyse other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if there is evidence of actions that are described there but not in the functional specification.

ADV_FSP.5.6C ***The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.***

ADV_FSP.5-8 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes all errors messages resulting from an invocation of each TSFI.

676 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.

677 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 (ADV_ARC, ADV_TDS) related that TSFI to determine if the description is complete and accurate.

678 The evaluator determines that, for each TSFI, the exact set of error messages that can be returned on invoking that interface can be determined. The evaluator reviews the evidence provided for the interface to determine if the set of errors seems complete. They cross-check this information with other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to ensure that there are no errors steaming from processing mentioned that are not included in the functional specification.

ADV_FSP.5-9 The evaluator ***shall examine*** the presentation of the TSFI to determine that it completely and accurately describes the meaning of all errors associated with each TSFI.

679 In order to determine accuracy, the evaluator must be able to understand meaning of the error. For example, if an interface returns a numeric code of 0, 1, or 2, the evaluator would not be able to understand the error if the functional specification only listed: "possible errors resulting from invocation of the *foo()* interface are 0, 1, or 2". Instead the evaluator checks to ensure that the errors are described such as: "possible errors resulting from invocation of the *foo()* interface are 0 (processing successful), 1 (file not found), or 2 (incorrect filename specification)".

680 In order to determine that the description of the errors due to invoking a TSFI is complete, the evaluator examines the rest of the interface description (parameter descriptions, actions, etc.) to determine if potential error conditions that might be caused by using such an interface are accounted for. The evaluator also checks other evidence provided for the evaluation (e.g., TOE design, security architecture description, operational user guidance, implementation representation) to see if error processing related to the TSFI is described there but is not described in the functional specification.

ADV_FSP.5.7C ***The functional specification shall describe all error messages that do not result from an invocation of a TSFI.***

ADV_FSP.5-10 The evaluator ***shall examine*** the functional specification to determine that it completely and accurately describes all errors messages that do not result from an invocation of any TSFI.

681 This work unit complements work unit ADV_FSP.5-8, which describes those error messages that result from an invocation of the TSFI. Taken together, these work units cover all error messages that might be generated by the TSF.

682 The evaluator assesses the completeness and accuracy of the functional specification by comparing its contents to instances of error message generation within the implementation representation. Most of these error messages will have already been covered by work unit ADV_FSP.5-8.

683 The error messages related to this work unit are typically those that are not expected to be generated, but are constructed as a matter of good programming practises. For example, a case statement that defines actions resulting from each of a list of cases may end with a final *else* statement to apply to anything that might not be expected; this practise ensures the TSF does not get into an undefined state. However, it is not expected that the path of execution would ever get to this *else* statement; therefore, any error message generation within this *else* statement would never be generated. Although it would not get generated, it must still be included in the functional specification.

ADV_FSP.5.8C ***The functional specification shall provide a rationale for each error message contained in the TSF implementation yet does not result from an invocation of a TSFI.***

ADV_FSP.5-11 The evaluator ***shall examine*** the functional specification to determine that it provides a rationale for each error message contained in the TSF implementation yet does not result from an invocation of a TSFI.

684 The evaluator ensures that every error message found under work unit ADV_FSP.5-10 contains a rationale describing why it cannot be invoked from the TSFI.

685 As was described in the previous work unit, this rationale might be as straightforward as the fact that the error message in question is provided for completeness of execution logic and that it is never expected to be generated. The evaluator ensures that the rationale for each such error message is logical.

ADV_FSP.5.9C ***The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.***

ADV_FSP.5-12 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.

686 The tracing is provided by the developer to serve as a guide to which SFRs are related to which TSFIs. This tracing can be as simple as a table; it is used as input to the evaluator for use in the following work units, in which the evaluator verifies its completeness and accuracy.

11.4.5.4 Action ADV_FSP.5.2E

ADV_FSP.5-13 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

687 To ensure that all SFRs are covered by the functional specification, as well as the test coverage analysis, the evaluator may build upon the developer's tracing (see ADV_FSP.5-12 a map between the TOE security functional requirements and the TSFI. Note that this map may have to be at a level of detail below the component or even element level of the requirements,

because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

688 For example, the FDP_ACC.1 component contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 assignment, and these ten rules were covered by three different TSFI, it would be inadequate for the evaluator to map FDP_ACC.1 to TSFI A, B, and C and claim they had completed the work unit. Instead, the evaluator would map FDP_ACC.1 (rule 1) to TSFI A; FDP_ACC.1 (rule 2) to TSFI B; etc. It might also be the case that the interface is a wrapper interface (e.g., IOCTL), in which case the mapping would need to be specific to certain set of parameters for a given interface.

689 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that they completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST. It is also important to note that since the parameters, actions, and error messages associated with TSFIs must be fully specified, the evaluator should be able to determine if all aspects of an SFR appear to be implemented at the interface level.

ADV_FSP.5-14 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

690 For each functional requirement in the ST that results in effects visible at the TSF boundary, the information in the associated TSFI for that requirement specifies the required functionality described by the requirement. For example, if the ST contains a requirement for access control lists, and the only TSFI that map to that requirement specify functionality for Unix-style protection bits, then the functionality specification is not accurate with respect to the requirements.

691 The evaluator must recognise that for requirements that have little or no manifestation at the TSF boundary (e.g., FDP_RIP) it is not expected that the evaluator completely map those requirements to the TSFI. The analysis for those requirements will be performed in the analysis for the TOE design (ADV_TDS) when included in the ST.

11.4.6 Evaluation of sub-activity (ADV_FSP.6)

692 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

11.5 Implementation representation (ADV_IMP)

11.5.1 Evaluation of sub-activity (ADV_IMP.1)

11.5.1.1 Objectives

693 The objective of this sub-activity is to determine that the implementation representation made available by the developer is suitable for use in other analysis activities; *suitability* is judged by its conformance to the requirements for this component.

11.5.1.2 Input

694 The evaluation evidence for this sub-activity is:

a) the implementation representation;

b) the documentation of the development tools, as resulting from ALC_TAT ;

c) TOE design description.

11.5.1.3 Application notes

695 The entire implementation representation is made available to ensure that analysis activities are not curtailed due to lack of information. This does not, however, imply that all of the representation is examined when the analysis activities are being performed. This is likely impractical in almost all cases, in addition to the fact that it most likely will not result in a higher-assurance TOE vs. targeted sampling of the implementation representation. For this sub-activity, this is even truer. It would not be productive for the evaluator to spend large amounts of time verifying the requirements for one portion of the implementation representation, and then use a different portion of the implementation representation in performing analysis for other work units. Therefore, the evaluator is encouraged to select the sample of the implementation representation from the areas of the TOE that will be of most interest during the analysis performed during work units from other families (e.g. ATE_IND, AVA_VAN and ADV_INT).

11.5.1.4 Action ADV_IMP.1.1E

ADV_IMP.1.1C ***The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.***

ADV_IMP.1-1 The evaluator ***shall check*** that the implementation representation defines the TSF to a level of detail such that the TSF can be generated without further design decisions.

696 Source code or hardware diagrams and/or IC hardware design language code or layout data that are used to build the actual hardware are examples of parts of an implementation representation. The evaluator samples the implementation representation to gain confidence that it is at the appropriate

level and not, for instance, a pseudo-code level which requires additional design decisions to be made. The evaluator is encouraged to perform a quick check when first looking at the implementation representation to assure themselves that the developer is on the right track. However, the evaluator is also encourage to perform the bulk of this check while working on other work units that call for examining the implementation; this will ensure the sample examined for this work unit is relevant.

ADV_IMP.1.2C ***The implementation representation shall be in the form used by the development personnel.***

ADV_IMP.1-2 The evaluator ***shall check*** that the implementation representation is in the form used by development personnel.

697 The implementation representation is manipulated by the developer in form that it suitable for transformation to the actual implementation. For instance, the developer may work with files containing source code, which is eventually compiled to become part of the TSF. The developer makes available the implementation representation in the form they use, so that the evaluator may use automated techniques in the analysis. This also increases the confidence that the implementation representation examined is actually the one used in the production of the TSF (as opposed to the case where it is supplied in an alternate presentation format, such as a word processor document). It should be noted that other forms of the implementation representation may also be used by the developer; these forms are supplied as well. The overall goal is to supply the evaluator with the information that will maximise the evaluator's analysis efforts.

698 The evaluator samples the implementation representation to gain confidence that it is the version that is usable by the developer. The sample is such that the evaluator has assurance that all areas of the implementation representation are in conformance with the requirement; however, a complete examination of the entire implementation representation is unnecessary.

699 Conventions in some forms of the implementation representation may make it difficult or impossible to determine from just the implementation representation itself what the actual result of the compilation or run-time interpretation will be. For example, compiler directives for C language compilers will cause the compiler to exclude or include entire portions of the code.

700 Some forms of the implementation representation may require additional information because they introduce significant barriers to understanding and analysis. Examples include shrouded source code or source code that has been obfuscated in other ways such that it prevents understanding and/or analysis. These forms of implementation representation typically result from by taking a version of the implementation representation that is used by the TOE developer and running a shrouding or obfuscation program on it. While the shrouded representation is what is compiled and may be closer to the implementation (in terms of structure) than the original, un-shrouded

representation, supplying such obfuscated code may cause significantly more time to be spent in analysis tasks involving the representation. When such forms of representation are created, the components require details on the shrouding tools/algorithms used so that the un-shrouded representation can be supplied, and the additional information can be used to gain confidence that the shrouding process does not compromise any security mechanisms.

701 The evaluator samples the implementation representation to gain confidence that all of the information needed to interpret the implementation representation has been supplied. Note that the tools are among those referenced by Tools and techniques (ALC_TAT) components. The evaluator is encouraged to perform a quick check when first looking at the implementation representation to assure themselves that the developer is on the right track. However, the evaluator is also encouraged to perform the bulk of this check while working on other work units that call for examining the implementation; this will ensure the sample examined for this work unit is relevant.

ADV_IMP.1.3C ***The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.***

ADV_IMP.1-3 The evaluator ***shall examine*** the mapping between the TOE design description and the sample of the implementation representation to determine that it is accurate.

702 The evaluator augments the determination of existence (specified in work unit ADV_IMP.1-1) by verifying the accuracy of a portion of the implementation representation and the TOE design description. For parts of the TOE design description that are interesting, the evaluator would verify the implementation representation accurately reflects the description provided in the TOE design description.

703 For example, the TOE design description might identify a login module that is used to identify and authenticate users. If user authentication is sufficiently significant, the evaluator would verify that the corresponding code in fact implements that service as described in the TOE design description. It might also be worthwhile to verify that the code accepts the parameters as described in the functional specification.

704 It is worth pointing out the developer must choose whether to perform the mapping for the entire implementation representation, thereby guaranteeing that the chosen sample will be covered, or waiting for the sample to be chosen before performing the mapping. The first option is likely more work, but may be completed before the evaluation begins. The second option is less work, but will produce a suspension of evaluation activity while the necessary evidence is being produced.

11.5.2 Evaluation of sub-activity (ADV_IMP.2)

705 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

11.6 TSF internals (ADV_INT)

11.6.1 Evaluation of sub-activity (ADV_INT.1)

11.6.1.1 Objectives

706 The objective of this sub-activity is to determine whether the defined subset of the TSF is designed and structured such that the likelihood of flaws is reduced and that maintenance can be more readily performed without the introduction of flaws.

11.6.1.2 Input

707 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE design description;

c) the implementation representation (if ADV_IMP is part of the claimed assurance);

d) the architectural description;

e) the documentation of the coding standards, as resulting from ALC_TAT.

11.6.1.3 Application notes

708 The role of the internals description is to provide evidence of the structure of the design and implementation of the TSF.

709 The structure of the design has two aspects: the constituent parts of the TSF and the procedures used to design the TSF. In cases where the TSF is designed in a manner consistent with the design represented by the TOE design (see ADV_TDS), the assessment of the TSF design is obvious. In cases where the design procedures (see ALC_TAT) are being followed, the assessment of the TSF design procedures is similarly obvious.

710 In cases where the TSF is implemented using procedure-based software, this structure is assessed on the basis of its *modularity*; the modules identified in the internals description are the same as the modules identified in the TOE design (TOE design (ADV_TDS)). A module consists of one or more source code files that cannot be decomposed into smaller compilable units.

711 The use of the assignment in this component levies stricter constraints on the identified subset of the TSF than on the remainder of the TSF. The TSF subset that is explicitly identified in the assignment ADV_INT.1.1D. While the entire TSF is to be designed using good engineering principles and result in a well-structured TSF, only the specified subset is specifically analysed for this characteristic. The evaluator determines that the developer's application of coding standards result in a TSF that is understandable.

712 The primary goal of this component is to ensure the TSF subset's implementation representation is understandable to facilitate maintenance and analysis (of both the developer and evaluator).

11.6.1.4 Action ADV_INT.1.1E

ADV_INT.1.1C ***The justification shall explain the characteristics used to judge the meaning of "well-structured".***

ADV_INT.1-1 The evaluator ***shall examine*** the justification to determine that it identifies the basis for determining whether the TSF is well-structured.

713 The evaluator verifies that the criteria for determining the characteristic of being well-structured are clearly defined in the justification. Acceptable criteria typically originate from industry standards for the technology discipline. For example, procedural software that executes linearly is traditionally viewed as well-structured if it adheres to software engineering programming practises, such as those defined in the IEEE Standard (*IEEE Std 610.12-1990*). For example, it would identify the criteria for the procedural software portions of the TSF subset:

a) the process used for modular decomposition

b) coding standards used in the development of the implementation

c) a description of the maximum acceptable level of intermodule coupling exhibited by the TSF subset

d) a description of the minimum acceptable level of cohesion exhibited the modules of the TSF subset

714 For other types of technologies used in the TOE - such as non-procedural software (e.g. object-oriented programming), widespread commodity hardware (e.g. PC microprocessors), and special-purpose hardware (e.g. smart-card processors) - the evaluation authority should be consulted for determining the adequacy of criteria for being "well-structured".

ADV_INT.1.2C ***The TSF internals description shall demonstrate that the assigned subset of the TSF is well-structured.***

ADV_INT.1-2 The evaluator ***shall check*** the TSF internals description to determine that it identifies the Assigned subset of the TSF.

715 This subset may be identified in terms of the internals of the TSF at any layer of abstraction. For example, it may be in terms of the structural elements of the TSF as identified in the TOE design (e.g. the audit subsystem), or in terms of the implementation (e.g. *encrypt.c* and *decrypt.c* files, or the 6227 IC chip).

716 It is insufficient to identify this subset in terms of the claimed SFRs (e.g. the portion of the TSF that provide anonymity as defined in FPR_ANO.2) because this does not indicate where to focus the analysis.

ADV_INT.1-3 The evaluator ***shall examine*** the TSF internals description to determine that it demonstrates that the assigned TSF subset is well-structured.

717 The evaluator examines the internals description to ensure that it provides a sound explanation of how the TSF subset meets the criteria from ADV_INT.1-1

718 For example, it would explain how the procedural software portions of the TSF subset meets the following:

a) that there is a one-to-one correspondence between the modules identified in the TSF subset and the modules described in the TOE design (ADV_TDS)

b) how the TSF design is a reflection of the modular decomposition process

c) a justification for all instances where the coding standards were not used or met

d) a justification for any coupling or cohesion outside the acceptable bounds

11.6.1.5 Action ADV_INT.1.2E

ADV_INT.1-4 The evaluator ***shall determine*** that the TOE design for the assigned TSF subset is well-structured.

719 The evaluator examines a sample of the TOE design to verify the accuracy of the justification. For example, a sample of the TOE design is analysed to determine its adherence to the design standards, etc. As with all areas where the evaluator performs activities on a subset the evaluator provides a justification of the sample size and scope

720 The description of the TOE's decomposition into subsystems and modules will make the argument that the TSF subset is well-structured self-evident. Verification that the procedures for structuring the TSF (as examined in ALC_TAT) are being followed will make it self-evident that the TSF subset is well-structured.

ADV_INT.1-5 The evaluator ***shall determine*** that the assigned TSF subset is well-structured.

721 If ADV_IMP is not part of the claimed assurance, then this work unit is not applicable and is therefore considered to be satisfied.

722 The evaluator examines a sample of the TSF subset to verify the accuracy of the internals description. For example, a sample of the procedural software portions of the TSF subset is analysed to determine its cohesion and coupling, its adherence to the coding standards, etc. As with all areas where the evaluator performs activities on a subset the evaluator provides a justification of the sample size and scope.

11.6.2 Evaluation of sub-activity (ADV_INT.2)

11.6.2.1 Objectives

723 The objective of this sub-activity is to determine whether the TSF is designed and structured such that the likelihood of flaws is reduced and that maintenance can be more readily performed without the introduction of flaws.

11.6.2.2 Input

724 The evaluation evidence for this sub-activity is:

a) the modular design description;

b) the implementation representation (if ADV_IMP is part of the claimed assurance));

c) the TSF internals description;

d) the documentation of the coding standards, as resulting from ALC_TAT.

11.6.2.3 Application notes

725 The role of the internals description is to provide evidence of the structure of the design and implementation of the TSF.

726 The structure of the design has two aspects: the constituent parts of the TSF and the procedures used to design the TSF. In cases where the TSF is designed in a manner consistent with the design represented by the TOE design (see ADV_TDS), the assessment of the TSF design is obvious. In cases where the design procedures (see ALC_TAT) are being followed, the assessment of the TSF design procedures is similarly obvious.

727 In cases where the TSF is implemented using procedure-based software, this structure is assessed on the basis of its *modularity*; the modules identified in the internals description are the same as the modules identified in the TOE design (TOE design (ADV_TDS)). A module consists of one or more source code files that cannot be decomposed into smaller compilable units.

728 The primary goal of this component is to ensure the TSF's implementation representation is understandable to facilitate maintenance and analysis (of both the developer and evaluator).

11.6.2.4 Action ADV_INT.2.1E

ADV_INT.2.1C ***The justification shall explain the characteristics used to judge the meaning of "well-structured".***

ADV_INT.2-1 The evaluator ***shall examine*** the justification to determine that it identifies the basis for determining whether the TSF is well-structured.

729 The evaluator verifies that the criteria for determining the characteristic of being well-structured are clearly defined in the justification. Acceptable criteria typically originate from industry standards for the technology discipline. For example, procedural software that executes linearly is traditionally viewed as well-structured if it adheres to software engineering programming practises, such as those defined in the IEEE Standard (*IEEE Std 610.12-1990*). For example, it would identify the criteria for the procedural software portions of the TSF:

a) the process used for modular decomposition

b) coding standards used in the development of the implementation

c) a description of the maximum acceptable level of intermodule coupling exhibited by the TSF

d) a description of the minimum acceptable level of cohesion exhibited the modules of the TSF

730 For other types of technologies used in the TOE - such as non-procedural software (e.g. object-oriented programming), widespread commodity hardware (e.g. PC microprocessors), and special-purpose hardware (e.g. smart-card processors) - the evaluation authority should be consulted for determining the adequacy of criteria for being “well-structured”.

ADV_INT.2.2C ***The TSF internals description shall demonstrate that the entire TSF is well-structured.***

ADV_INT.2-2 The evaluator ***shall examine*** the TSF internals description to determine that it demonstrates that the TSF is well-structured.

731 The evaluator examines the internals description to ensure that it provides a sound explanation of how the TSF meets the criteria from ADV_INT.2-1

732 For example, it would explain how the procedural software portions of the TSF meet the following:

a) that there is a one-to-one correspondence between the modules identified in the TSF and the modules described in the TOE design (ADV_TDS)

b) how the TSF design is a reflection of the modular decomposition process

c) a justification for all instances where the coding standards were not used or met

d) a justification for any coupling or cohesion outside the acceptable bounds

11.6.2.5 Action ADV_INT.2.2E

ADV_INT.2-3 The evaluator ***shall determine*** that the TOE design is well-structured.

733 The evaluator examines the TOE design of a sample of the TSF to verify the accuracy of the justification. For example, a sample of the TOE design is analysed to determine its adherence to the design standards, etc. As with all areas where the evaluator performs activities on a subset the evaluator provides a justification of the sample size and scope

734 The description of the TOE's decomposition into subsystems and modules will make the argument that the TSF subset is well-structured self-evident. Verification that the procedures for structuring the TSF (as examined in ALC_TAT) are being followed will make it self-evident that the TSF subset is well-structured.

ADV_INT.2-4 The evaluator ***shall determine*** that the TSF is well-structured.

735 If ADV_IMP is not part of the claimed assurance, then this work unit is not applicable and is therefore considered to be satisfied.

736 The evaluator examines a sample of the TSF to verify the accuracy of the internals description. For example, a sample of the procedural software portions of the TSF is analysed to determine its cohesion and coupling, its adherence to the coding standards, etc. As with all areas where the evaluator performs activities on a subset the evaluator provides a justification of the sample size and scope.

11.6.3 Evaluation of sub-activity (ADV_INT.3)

11.6.3.1 Input

737 The evaluation evidence for this sub-activity is:

a) the modular design description;

b) the implementation representation;

c) the TSF internals description;

d) the documentation of the coding standards, as resulting from ALC_TAT.

11.6.3.2 Action ADV_INT.3.1E

11.6.3.3 Action ADV_INT.3.2E

11.7 Security policy modelling (ADV_SPM)

11.7.1 Evaluation of sub-activity (ADV_SPM.1)

738 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

11.8 TOE design (ADV_TDS)

11.8.1 Evaluation of sub-activity (ADV_TDS.1)

11.8.1.1 Input

739 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) security architecture description;

d) the TOE design.

11.8.1.2 Action ADV_TDS.1.1E

ADV_TDS.1.1C ***The design shall describe the structure of the TOE in terms of subsystems.***

ADV_TDS.1-1 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is described in terms of subsystems.

740 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the TOE will be used as input to work unit ADV_TDS.1-2, where the parts of the TOE that make up the TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

741 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and modules) Depending upon the complexity of the TOE, its design may be described in terms of subsystems and modules, as described in CC Part 3 Annex A.4, ADV_TDS: Subsystems and Modules. At this level of assurance, the decomposition only need be at the “subsystem” level.

742 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST, operator user guidance) to determine that the description of the TOE in such evidence is consistent with the description contained in the TOE design.

ADV_TDS.1.2C ***The design shall identify all subsystems of the TSF.***

ADV_TDS.1-2 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are identified.

743 In work unit ADV_TDS.1-1 all of the subsystems of the TOE were identified, and a determination made that the non-TSF subsystems were correctly characterised. Building on that work, the subsystems that were not characterised as non-TSF subsystems should be precisely identified. The evaluator determines that, of the hardware and software installed and configured according to the Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one that is part of the TSF, or one that is not.

ADV_TDS.1.3C ***The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.***

ADV_TDS.1-3 The evaluator ***shall examine*** the TOE design to determine that each non-SFR-enforcing subsystem of the TSF is described such that the evaluator can determine that the subsystem is non-SFR-enforcing.

744 Non-SFR-enforcing subsystems do not need to be described in detail as to how they function in the system. However, the evaluator makes a determination, based on the evidence provided by the developer, that the subsystems that do not have high-level descriptions are non-SFR-enforcing (that is, either SFR-supporting or SFR-non-interfering). Note that if the developer provides a uniform level of detailed documentation then this work unit will be largely satisfied, since the point of categorising the subsystems is to allow the developer to provide less information for non-SFR-enforcing subsystems than for SFR-enforcing subsystems.

745 An SFR-supporting subsystem is one that is depended on by an SFR-enforcing subsystem in order to implement an SFR, but does not play as direct a role as an SFR-supporting requirement. An SFR-non-interfering subsystem is one that is not depended upon, in either a supporting or enforcing role, to implement an SFR.

ADV_TDS.1.4C ***The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.***

ADV_TDS.1-4 The evaluator ***shall examine*** the TOE design to determine that it provides a complete, accurate, and high-level description of the SFR-enforcing behaviour of the SFR-enforcing subsystems.

746 The developer may designate subsystems 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 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to determine that the subsystems 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 subsystem.

747 SFR-enforcing behaviour refers to *how* a subsystem provides the functionality that implements an SFR. A high-level description need not refer to specific data structures (although it may), but instead talks about more general data flow, message flow, and control relationships within a subsystem. The goal of these descriptions is to give the evaluator enough information to understand *how* the SFR-enforcing behaviour is achieved. Note that the evaluator should find unacceptable asserts of SFR-enforcement in the TOE design documentation for this work unit. It should be noted that it is the evaluator's determination with respect to what “high-level” means for a

particular TOE, and the evaluator obtains enough information from the developer to make a sound verdict for this work unit.

748 To determine completeness and accuracy, the evaluator examines other information available (e.g., functional specification, security architecture description, implementation representation). Descriptions of functionality in these documents should be consistent with what is provided for evidence for this work unit

ADV_TDS.1.5C ***The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.***

ADV_TDS.1-5 The evaluator ***shall examine*** the TOE design to determine that interactions between the subsystems of the TSF are described.

749 The goal of describing the interactions between the SFR-enforcing subsystems and other subsystems is to help provide the reader a better understanding of how the TSF performs it functions. These interactions do not need to be characterised at the implementation level (e.g., parameters passed from one routine in a subsystem to a routine in a different subsystem; global variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a particular subsystem that are going to be used by another subsystem should be covered in this discussion. Any control relationships between subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the subsystem that actually implements these rules) should also be described.

750 The evaluators should use their own judgement in assessing the completeness of the description. If the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for instance, in examining the descriptions of subsystem behaviour) that do not appear to be described, the evaluator ensures that this information is provided by the developer. However, if the evaluator can determine that interactions among a particular set of subsystems, while incompletely described by the developer, will not aid in understanding the overall functionality nor security functionality provided by the TSF, then the evaluator may choose to consider the description sufficient, and not pursue completeness for its own sake.

ADV_TDS.1.6C ***The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.***

ADV_TDS.1-6 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate mapping from the TSFI described in the functional specification to the subsystems of the TSF described in the TOE design.

751 The subsystems described in the TOE design provide a description of how the TSF works at a detailed level for SFR-enforcing portions of the TSF, and at a higher level for other portions of the TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the

developer identifies the subsystem that is initially involved when an operation is requested at the TSFI, and identify the various subsystems that are primarily responsible for implementing the functionality. Note that a complete "call tree" for each TSFI is not required for this work unit.

752 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at least one subsystem. The verification of accuracy is more complex.

753 The first aspect of accuracy is that each TSFI is mapped to a subsystem at the TSF boundary. This determination can be made by reviewing the subsystem description and interactions, and from this information determining its place in the architecture. The next aspect of accuracy is that the mapping makes sense. For instance, mapping a TSFI dealing with access control to a subsystem that checks passwords is not accurate. The evaluator should again use judgement in making this determination. The goal is that this information aids the evaluator in understanding the system and implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the TSF. The bulk of the assessment of whether the SFRs are described accurately by the subsystems is performed in other work units.

11.8.1.3 Action ADV_TDS.1.2E

ADV_TDS.1-7 The evaluator ***shall examine*** the TOE security functional requirements and the TOE design, to determine that all ST security functional requirements are covered by the TOE design.

754 The evaluator may construct a map between the TOE security functional requirements and the TOE design. This map will likely be from a functional requirement to a set of subsystems. Note that this map may have to be at a level of detail below the subsystem or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

755 For example, the FDP_ACC.1 Subset access control subsystem contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 Subset access control assignment, and these ten rules were implemented in specific places within fifteen modules, it would be inadequate for the evaluator to map FDP_ACC.1 Subset access control to one subsystem and claim the work unit had been completed. Instead, the evaluator would map FDP_ACC.1 Subset access control (rule 1) to subsystem A, behaviours x, y, and z; FDP_ACC.1 Subset access control (rule 2) to subsystem A, behaviours x, p, and q; etc.

ADV_TDS.1-8 The evaluator ***shall examine*** the TOE design to determine that it is an accurate instantiation of all security functional requirements.

756 The evaluator ensures that each security requirement listed in the TOE security functional requirements section of the ST has a corresponding design description in the TOE design that accurately details how the TSF meets that requirement. This requires that the evaluator identify a collection

of subsystems that are responsible for implementing a given functional requirement, and then examine those subsystems to understand how the requirement is implemented. Finally, the evaluator would assess whether the requirement was accurately implemented.

757 As an example, if the ST requirements specified a role-based access control mechanism, the evaluator would first identify the subsystems that contribute to this mechanism's implementation. This could be done by in-depth knowledge or understanding of the TOE design or by work done in the previous work unit. Note that this trace is only to identify the subsystems, and is not the complete analysis.

758 The next step would be to understand what mechanism the subsystems implemented. For instance, if the design described an implementation of access control based on UNIX-style protection bits, the design would not be an accurate instantiation of those access control requirements present in the ST example used above. If the evaluator could not determine that the mechanism was accurately implemented because of a lack of detail, the evaluator would have to assess whether all of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for those subsystems.

11.8.2 Evaluation of sub-activity (ADV_TDS.2)

11.8.2.1 Input

759 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) security architecture description;

d) the TOE design.

11.8.2.2 Action ADV_TDS.2.1E

ADV_TDS.2.1C ***The design shall describe the structure of the TOE in terms of subsystems.***

ADV_TDS.2-1 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is described in terms of subsystems.

760 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the TOE will be used as input to work unit ADV_TDS.2-2, where the parts of the TOE that make up the TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

761 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and modules) Depending upon the complexity of the TOE, its design may be described in terms of subsystems and modules, as described

in CC Part 3 Annex A.4, ADV_TDS: Subsystems and Modules. At this level of assurance, the decomposition only need be at the “subsystem” level.

762 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST, operator user guidance) to determine that the description of the TOE in such evidence is consistent with the description contained in the TOE design.

ADV_TDS.2.2C ***The design shall identify all subsystems of the TSF.***

ADV_TDS.2-2 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are identified.

763 In work unit ADV_TDS.2-1 all of the subsystems of the TOE were identified, and a determination made that the non-TSF subsystems were correctly characterised. Building on that work, the subsystems that were not characterised as non-TSF subsystems should be precisely identified. The evaluator determines that, of the hardware and software installed and configured according to the Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one that is part of the TSF, or one that is not.

ADV_TDS.2.3C ***The design shall describe the behaviour of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.***

ADV_TDS.2-3 The evaluator ***shall examine*** the TOE design to determine that each SFR-non-interfering subsystem of the TSF is described such that the evaluator can determine that the subsystem is SFR-non-interfering.

764 SFR-non-interfering subsystems do not need to be described in detail as to how they function in the system. However, the evaluator makes a determination, based on the evidence provided by the developer, that the subsystems that do not have detailed descriptions are SFR-non-interfering. Note that if the developer provides a uniform level of detailed documentation then this work unit will be largely satisfied, since the point of categorising the subsystems is to allow the developer to provide less information for SFR-non-interfering subsystems than for SFR-enforcing and SFR-supporting subsystems.

765 An SFR-non-interfering subsystem is one on which the SFR-enforcing and SFR-supporting subsystems have no dependence; that is, they play no role in implementing SFR functionality.

ADV_TDS.2.4C ***The design shall describe the SFR-enforcing behaviour of the SFR-enforcing subsystems.***

ADV_TDS.2-4 The evaluator ***shall examine*** the TOE design to determine that it provides a complete, accurate, and detailed description of the SFR-enforcing behaviour of the SFR-enforcing subsystems.

766 The developer may designate subsystems 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 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to determine that the subsystems 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 subsystem.

767 SFR-enforcing behaviour refers to *how* a subsystem provides the functionality that implements an SFR. While not at the level of an algorithmic description, a detailed description of behaviour typically discusses how the functionality is provided in terms of what key data and data structures are, what control relationships exist within a subsystem, and how these elements work together to provide the SFR-enforcing behaviour. Such a description also references SFR-supporting behaviour, which the evaluator should consider in performing subsequent work units.

768 To determine completeness and accuracy, the evaluator examines other information available (e.g., functional specification, security architecture description, implementation representation). Descriptions of functionality in these documents should be consistent with what is provided for evidence for this work unit

ADV_TDS.2.5C ***The design shall summarise the non-SFR-enforcing behaviour of the SFR-enforcing subsystems.***

ADV_TDS.2-5 The evaluator ***shall examine*** the TOE design to determine that it provides a complete and accurate high-level description of the non-SFR-enforcing behaviour of the SFR-enforcing subsystems.

769 The developer may designate subsystems 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 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to determine that the subsystems 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 subsystem.

770 In contrast to the previous work unit, this work unit calls for the evaluator to assess the information provided for SFR-enforcing subsystems that is non-SFR-enforcing. The goal of this assessment is two-fold. First, it should provide the evaluator greater understanding of the way each subsystem works. Second, the evaluator determines that all SFR-enforcing behaviour exhibited by a subsystem has been described. Unlike the previous work unit, the information provided for non-SFR-enforcing behaviour does not have to

be as detailed as that provided by the SFR-enforcing behaviour. For example, data structures or data items that do not pertain to SFR-enforcing functionality will likely not need to be described in detail, if at all. It is the evaluator's determination, however, with respect to what “high-level” means for a particular TOE, and the evaluator obtains enough information from the developer (even if it turns out to be equivalent to information provided for the parts of the subsystem that are SFR-enforcing) to make a sound verdict for this work unit.

771 The evaluator is cautioned, however, that “perfect” assurance is not a goal nor required by this work unit, so judgement will have to be exercised in determine the amount and composition of the evidence required to make a verdict on this work unit.

772 To determine completeness and accuracy, the evaluator examines other information available (e.g., functional specification, security architecture description, implementation representation). Descriptions of functionality in these documents should be consistent with what is provided for evidence for this work unit. In particular, the functional specification should be used to determine that the behaviour required to implement the TSF Interfaces described by the functional specification are completely described by the subsystem, since the behaviour will either be SFR-enforcing or non-SFR-enforcing.

ADV_TDS.2.6C ***The design shall summarise the behaviour of the SFR-supporting subsystems.***

ADV_TDS.2-6 The evaluator ***shall examine*** the TOE design to determine that it provides a complete and accurate high-level description of the behaviour of the SFR-supporting subsystems.

773 The developer may designate subsystems 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 subsystems have been categorised by the developer or not, it is the evaluator's responsibility to determine that the subsystems 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 subsystem.

774 In contrast to the previous two work units, this work unit calls for the developer to provide (and the evaluator to assess) information about SFR supporting subsystems. Such subsystems should be referenced by the descriptions of the SFR-enforcing subsystems, as well as by the descriptions of interactions in work unit ADV_TDS.2-7. The goal of evaluator's assessment, like that for the previous work unit, is two-fold. First, it should provide the evaluator with an understanding of the way each SFR-supporting subsystem works. Second, the evaluator determines that the behaviour is described in enough detail so that the way in which the subsystem supports

the SFR-enforcing behaviour is clear, and that the behaviour is not itself SFR-enforcing. The information provided for SFR-supporting subsystem's behaviour does not have to be as detailed as that provided by the SFR-enforcing behaviour. For example, data structures or data items that do not pertain to SFR-enforcing functionality will likely not need to be described in detail, if at all. It is the evaluator's determination, however, with respect to what "high-level" means for a particular TOE, and the evaluator obtains enough information from the developer (even if it turns out to be equivalent to information provided for the parts of the subsystem that are SFR-enforcing) to make a sound verdict for this work unit.

775 The evaluator is cautions, however, that "perfect" assurance is not a goal nor required by this work unit, so judgement will have to be exercised in determine the amount and composition of the evidence required to make a verdict on this work unit.

776 To determine completeness and accuracy, the evaluator examines other information available (e.g., functional specification, security architecture description, implementation representation). Descriptions of functionality in these documents should be consistent with what is provided for evidence for this work unit. In particular, the functional specification should be used to determine that the behaviour required to implement the TSF Interfaces described by the functional specification are completely described by the subsystem.

ADV_TDS.2.7C ***The design shall provide a description of the interactions among all subsystems of the TSF.***

ADV_TDS.2-7 The evaluator ***shall examine*** the TOE design to determine that interactions between the subsystems of the TSF are described.

777 The goal of describing the interactions between the subsystems is to help provide the reader a better understanding of how the TSF performs it functions. These interactions do not need to be characterised at the implementation level (e.g., parameters passed from one routine in a subsystem to a routine in a different subsystem; global variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a particular subsystem that are going to be used by another subsystem should be covered in this discussion. Any control relationships between subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the subsystem that actually implements these rules) should also be described.

778 It should be noted while the developer should characterise all interactions between subsystems, the evaluators should use their own judgement in assessing the completeness of the description. If the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for instance, in examining the descriptions of subsystem behaviour) that do not appear to be described, the evaluator ensures that this information is provided by the developer. However, if the evaluator can determine that interactions among a particular set of subsystems, while incompletely described by the developer,

will not aid in understanding the overall functionality nor security functionality provided by the TSF, then the evaluator may choose to consider the description sufficient, and not pursue completeness for its own sake.

ADV_TDS.2.8C ***The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.***

ADV_TDS.2-8 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate mapping from the TSFI described in the functional specification to the subsystems of the TSF described in the TOE design.

779 The subsystems described in the TOE design provide a description of how the TSF works at a detailed level for SFR-enforcing portions of the TSF, and at a higher level for other portions of the TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the developer identifies the subsystem that is initially involved when an operation is requested at the TSFI, and identify the various subsystems that are primarily responsible for implementing the functionality. Note that a complete "call tree" for each TSFI is not required for this work unit.

780 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at least one subsystem. The verification of accuracy is more complex.

781 The first aspect of accuracy is that each TSFI is mapped to a subsystem at the TSF boundary. This determination can be made by reviewing the subsystem description and interactions, and from this information determining its place in the architecture. The next aspect of accuracy is that the mapping makes sense. For instance, mapping a TSFI dealing with access control to a subsystem that checks passwords is not accurate. The evaluator should again use judgement in making this determination. The goal is that this information aids the evaluator in understanding the system and implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the TSF. The bulk of the assessment of whether the SFRs are described accurately by the subsystems is performed in other work units.

11.8.2.3 Action ADV_TDS.2.2E

ADV_TDS.2-9 The evaluator ***shall examine*** the TOE security functional requirements and the TOE design, to determine that all ST security functional requirements are covered by the TOE design.

782 The evaluator may construct a map between the TOE security functional requirements and the TOE design. This map will likely be from a functional requirement to a set of subsystems. Note that this map may have to be at a level of detail below the subsystem or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

783 For example, the FDP_ACC.1 Subset access control subsystem contains an element with assignments. If the ST contained, for instance, ten rules in the

FDP_ACC.1 Subset access control assignment, and these ten rules were implemented in specific places within fifteen modules, it would be inadequate for the evaluator to map FDP_ACC.1 Subset access control to one subsystem and claim the work unit had been completed. Instead, the evaluator would map FDP_ACC.1 Subset access control (rule 1) to subsystem A, behaviours x, y, and z; FDP_ACC.1 Subset access control (rule 2) to subsystem A, behaviours x, p, and q; etc.

ADV_TDS.2-10 The evaluator ***shall examine*** the TOE design to determine that it is an accurate instantiation of all security functional requirements.

784 The evaluator ensures that each security requirement listed in the TOE security functional requirements section of the ST has a corresponding design description in the TOE design that accurately details how the TSF meets that requirement. This requires that the evaluator identify a collection of subsystems that are responsible for implementing a given functional requirement, and then examine those subsystems to understand how the requirement is implemented. Finally, the evaluator would assess whether the requirement was accurately implemented.

785 As an example, if the ST requirements specified a role-based access control mechanism, the evaluator would first identify the subsystems that contribute to this mechanism's implementation. This could be done by in-depth knowledge or understanding of the TOE design or by work done in the previous work unit. Note that this trace is only to identify the subsystems, and is not the complete analysis.

786 The next step would be to understand what mechanism the subsystems implemented. For instance, if the design described an implementation of access control based on UNIX-style protection bits, the design would not be an accurate instantiation of those access control requirements present in the ST example used above. If the evaluator could not determine that the mechanism was accurately implemented because of a lack of detail, the evaluator would have to assess whether all of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for those subsystems.

11.8.3 Evaluation of sub-activity (ADV_TDS.3)

11.8.3.1 Objectives

787 The objective of this sub-activity is to determine whether the TOE design provides a description of the TOE in terms of subsystems sufficient to determine the TSF boundary, and provides a description of the TSF internals in terms of modules (and optionally higher-level abstractions). It provides a detailed description of the SFR-enforcing modules and enough information about the SFR-supporting and SFR-non-interfering modules for the evaluator to determine that the SFRs are completely and accurately implemented; as such, the TOE design provides an explanation of the implementation representation.

11.8.3.2 Input

788 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) security architecture description;

d) the TOE design.

11.8.3.3 Application notes

789 There are three types of activity that the evaluator must undertake with respect to the TOE design. First, the evaluator determines that the TSF boundary has been adequately described. Second, the evaluator determines that the developer has provided documentation that conforms to the content and presentation requirements for this subsystem, and that is consistent with other documentation provided for the TOE. Finally, the evaluator must analyse the design information provided for the SFR-enforcing modules (at a detailed level) and the non-SFR-enforcing modules (at a less detailed level) to understand how the system is implemented, and with that knowledge ensure that the TSFI in the functional specification are adequately described, and that the test information adequately tests the TSF (done in the Class ATE: Tests work units).

790 It is important to note that while the developer is obligated to provide a complete description of the TSF (although SFR-enforcing modules will have more detail than the non-SFR-enforcing modules), the evaluator is expected to use their judgement in performing their analysis. While the evaluator is expected to look at every module, the detail to which they examine each module may vary. The evaluator analyses each module in order to gain enough understanding to determine the effect of the functionality of the module on the security of the system, and the depth to which they need to analyse the module may vary depending on the module's role in the system. An important aspect of this analysis is that the evaluator should use the other documentation provided (TSS, functional specification, security architecture description, and the TSF internal document) in order to determine that the functionality that is described is correct, and that the implicit designation of non-SFR-enforcing modules (see below) is supported by their role in the system architecture.

791 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.

11.8.3.4 Action ADV_TDS.3.1E

ADV_TDS.3.1C ***The design shall describe the structure of the TOE in terms of subsystems.***

ADV_TDS.3-1 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is described in terms of subsystems.

792 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the TOE will be used as input to work unit ADV_TDS.3-2, where the parts of the TOE that make up the TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

793 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and modules) Depending upon the complexity of the TOE, its design may be described in terms of subsystems and modules, as described in CC Part 3 Annex A.4, ADV_TDS: Subsystems and Modules. For a very simple TOE that can be described solely at the "module" level (see ADV_TDS.3-2), this work unit is not applicable and therefore considered to be satisfied.

794 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST, operator user guidance) to determine that the description of the TOE in such evidence is consistent with the description contained in the TOE design.

ADV_TDS.3.2C ***The design shall describe the TSF in terms of modules.***

ADV_TDS.3-2 The evaluator ***shall examine*** the TOE design to determine that the entire TSF is described in terms of modules.

795 The evaluator will examine the modules for specific properties in other work units; in this work unit the evaluator determines that the modular description covers the entire TSF, and not just a portion of the TSF. The evaluator uses other evidence provided for the evaluation (e.g., functional specification, architectural description) in making this determination. For example, if the functional specification contains interfaces to functionality that does not appear to be described in the TOE design description, it may be the case that a portion of the TSF has not been included appropriately. Making this determination will likely be an iterative process, where as more analysis is done on the other evidence, more confidence can be gained with respect to the completeness of the documentation.

796 Unlike subsystems, modules describe the implementation in a level of detail that can serve as a guide to reviewing the implementation representation. A description of a module should be such that one could create an implementation of the module from the description, and the resulting implementation would be 1) identical to the actual TSF implementation in terms of the interfaces presented and used by the module, and 2)

algorithmically identical to the TSF module. For instance, RFC 793 provides a high-level description of the TCP protocol. It is necessarily implementation independent. While it provides a wealth of detail, it is ***not*** a suitable design description because it is not specific to an implementation. An actual implementation can add to the protocol specified in the RFC, and implementation choices (for instance, the use of global data vs. local data in various parts of the implementation) may have an impact on the analysis that is performed. The design description of the TCP module would list the interfaces presented by the implementation (rather than just those defined in RFC 793), as well as an algorithm description of the processing associated with the modules implementing TCP (assuming it was part of the TSF).

ADV_TDS.3.3C ***The design shall identify all subsystems of the TSF.***

ADV_TDS.3-3 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are identified.

797 If the design is presented solely in terms of modules, then subsystems in these requirements are equivalent to modules and the activity should be performed at the module level.

798 In work unit ADV_TDS.3-1 all of the subsystems of the TOE were identified, and a determination made that the non-TSF subsystems were correctly characterised. Building on that work, the subsystems that were not characterised as non-TSF subsystems should be precisely identified. The evaluator determines that, of the hardware and software installed and configured according to the Preparative procedures (AGD_PRE) guidance, each subsystem has been accounted for as either one that is part of the TSF, or one that is not.

ADV_TDS.3.4C ***The design shall provide a description of each subsystem of the TSF.***

ADV_TDS.3-4 The evaluator ***shall examine*** the TOE design to determine that each subsystem of the TSF describes its role in the enforcement of SFRs described in the ST.

799 If the design is presented solely in terms of modules, then this work unit will be considered satisfied by the assessment done in subsequent work units; no explicit action on the part of the evaluator is necessary in this case.

800 On systems that are complex enough to warrant a subsystem-level description of the TSF in addition to the modular description, the goal of the subsystem-level description is to give the evaluator context for the modular description that follows. Therefore, the evaluator ensures that the subsystem-level description contains a description of how the security functional requirements are achieved in the design, but at a level of abstraction above the modular description. This description should discuss the mechanisms used at a level that is aligned with the module description; this will provide the evaluators the road map needed to intelligently assess the information contained in the module description. A well-written set of subsystem descriptions will help guide the evaluator in determining the modules that are

most important to examine, thus focusing the evaluation activity on the portions of the TSF that have the most relevance with respect to the enforcement of the SFRs.

801 The evaluator ensures that all subsystems of the TSF have a description. While the description should focus on the role that the subsystem plays in enforcing or supporting the implementation of the SFRs, enough information must be present so that a context for understanding the SFR-related functionality is provided.

ADV_TDS.3.5C ***The design shall provide a description of the interactions among all subsystems of the TSF.***

ADV_TDS.3-5 The evaluator ***shall examine*** the TOE design to determine that interactions between the subsystems of the TSF are described.

802 If the design is presented solely in terms of modules, then this work unit will be considered satisfied by the assessment done in subsequent work units; no explicit action on the part of the evaluator is necessary in this case.

803 On systems that are complex enough to warrant a subsystem-level description of the TSF in addition to the modular description, the goal of describing the interactions between the subsystems is to help provide the reader a better understanding of how the TSF performs it functions. These interactions do not need to be characterised at the implementation level (e.g., parameters passed from one routine in a subsystem to a routine in a different subsystem; global variables; hardware signals (e.g., interrupts) from a hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a particular subsystem that are going to be used by another subsystem should be covered in this discussion. Any control relationships between subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the subsystem that actually implements these rules) should also be described.

804 It should be noted while the developer should characterise all interactions between subsystems, the evaluators should use their own judgement in assessing the completeness of the description. If the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for instance, in examining the module-level documentation) that do not appear to be described, the evaluator ensures that this information is provided by the developer. However, if the evaluator can determine that interactions among a particular set of subsystems, while incompletely described by the developer, and a complete description will not aid in understanding the overall functionality nor security functionality provided by the TSF, then the evaluator may choose to consider the description sufficient, and not pursue completeness for its own sake.

ADV_TDS.3.6C ***The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.***

ADV_TDS.3-6 The evaluator ***shall examine*** the TOE design to determine that the mapping between the subsystems of the TSF to the modules of the TSF is complete.

805 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

806 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. To determine completeness, the evaluator examines each mapping and determines that all subsystems map to at least one module, and that all modules map to exactly one subsystem.

ADV_TDS.3-7 The evaluator ***shall examine*** the TOE design to determine that the mapping between the subsystems of the TSF to the modules of the TSF is accurate.

807 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

808 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 mis-understandings 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.3.7C ***The design shall describe each SFR-enforcing module in terms of its purpose.***

ADV_TDS.3-8 The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of each SFR-enforcing module is complete and accurate.

809 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.

810 The purpose of a module provides a description indicating what function the module is fulfilling. A word of caution to 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).

ADV_TDS.3.8C ***The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, and called interfaces to other modules.***

ADV_TDS.3-9 The evaluator ***shall examine*** the TOE design to determine that the description of the interfaces presented by each SFR-enforcing module contain an accurate and complete description of the SFR-related parameters, the invocation conventions for each interface, and any values returned directly by the interface.

811 The SFR-related interfaces of a module are those interfaces used by other modules as a means to invoke the SFR-related operations provided, and to provide inputs to or receive outputs from the module. The purpose in the specification of these interfaces is to permit the exercise of them during testing. Inter-module interfaces that are not SFR-related need not be specified or described, since they are not a factor in testing. Likewise, other internal interfaces that are not a factor in traversing SFR-related paths of execution (such as those internal paths that are fixed).

812 SFR-related interfaces are described in terms of how they are invoked, and any values that are returned. This description would include a list of SFR-related parameters, and descriptions of these parameters. Note that global data would also be considered parameters if used by the module (either as inputs or outputs) when invoked. If a parameter were expected to take on a set of values (e.g., a “flag” parameter), the complete set of values the parameter could take on that would have an effect on module processing would be specified. Likewise, parameters representing data structures are described such that each field of the data structure is identified and described. Note that different programming languages may have additional “interfaces” that would be non-obvious; an example would be operator/function overloading in C++. This “implicit interface” in the class description would also be described as part of the low-level TOE design. Note that although a module could present only one interface, it is more common that a module presents a small set of related interfaces.

813 In terms of the assessment of parameters (inputs and outputs) to a module, any use of global data must also be considered. A module “uses” global data if it either reads or writes the data. In order to assure the description of such parameters (if used) is complete, the evaluator uses other information provided about the module in the TOE design (interfaces, algorithmic description, etc.), as well as the description of the particular set of global data assessed in work unit ADV_TDS.3-8. For instance, the evaluator could first determine the processing the module performs by examining its function and interfaces presented (particularly the parameters of the interfaces). They could then check to see if the processing appears to “touch” any of the global data areas identified in the TOE design. The evaluator then determines that, for each global data area that appears to be “touched”, that global data area is listed as a means of input or output by the module the evaluator is examining.

814 Invocation conventions are a programming-reference-type description that one could use to correctly invoke a module's interface if one were writing a program to make use of the module's functionality through that interface. This includes necessary inputs and outputs, including any set-up that may need to be performed with respect to global variables.

815 Values returned through the interface refer to values that are either passed through parameters or messages; values that the function call itself returns in the style of a “C” program function call; or values passed through global means (such as certain error routines in *ix-style operating systems).

816 In order to assure the description is complete, the evaluator uses other information provided about the module in the TOE design (e.g., algorithmic description, global data used) to ensure that it appears all data necessary for performing the functions of the module is presented to the module, and that any values that other modules expect the module under examination to provide are identified as being returned by the module. The evaluator determines accuracy by ensuring that the description of the processing matches the information listed as being passed to or from an interface.

817 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 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.3.10C element, which maps the TSFI in the functional specification to the modules of the TSF.

ADV_TDS.3.9C ***The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.***

ADV_TDS.3-10 The evaluator ***shall examine*** the TOE design to determine that non-SFR-supporting modules are correctly categorised.

818 In the cases where the developer has provided different amounts of information for different modules, an implicit categorisation has been done. That is, modules (for instance) with detail presented on their SFR-related interfaces (see ADV_TDS.3.10C) are candidate SFR-enforcing modules, although examination by the evaluator may lead to a determination that some set of them are non-SFR-enforcing. Those with only a description of their purpose and interaction with other modules (for instance) are "implicitly categorised" as non-SFR-enforcing.

819 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 non-SFR-enforcing and the evaluation information about other modules (in the TOE design, the functional specification, the security architecture description, and the administrative guidance), whether the module is indeed non-SFR-enforcing. At this level of assurance some error should be tolerated; the evaluator does not have to be absolutely sure that a given module is non-SFR-enforcing, even though it is labelled as such. However, if the evidence provided indicates that a non-SFR-enforcing 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 non-SFR-enforcing). 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.

820 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 non-SFR-enforcing). 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.

821 A final example 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 non-SFR-enforcing). 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 be to 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.

ADV_TDS.3-11 The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of each non-SFR-supporting module is complete and accurate.

822 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.

823 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 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.3.10C element, which maps the TSFI in the functional specification to the modules of the TSF.

ADV_TDS.3-12 The evaluator ***shall examine*** the TOE design to determine that the description of a non-SFR-supporting module's interaction with other modules is complete and accurate.

824 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.

825 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 non-SFR-enforcing

modules in the overall TOE design. Understanding of this role will aid the evaluator in performing work unit ADV_TDS.3-15.

826 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; the interactions should describe what the module depends on from other modules in order to accomplish this function.

827 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 that the interactions are accurately and completely described.

ADV_TDS.3.10C ***The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.***

ADV_TDS.3-13 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate mapping from the TSFI described in the functional specification to the modules of the TSF described in the TOE design.

828 The modules described in the TOE design provide a description of the implementation of the TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the developer identifies the module that is initially invoked when an operation is requested at the TSFI, and identify the chain of modules invoked up to the module that is primarily responsible for implementing the functionality. However, a complete call tree for each TSFI is not required for this work unit. The cases in which more than one module would have to be identified are where there are “entry point” modules or wrapper modules that have no functionality other than conditioning inputs or de-multiplexing an input. Mapping to one of these modules would not provide any useful information to the evaluator.

829 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at least one module. The verification of accuracy is more complex.

830 The first aspect of accuracy is that each TSFI is mapped to a module at the TSF boundary. This determination can be made by reviewing the module description and its interfaces/interactions. The next aspect of accuracy is that each TSFI identifies a chain of modules between the initial module identified and a module that is primarily responsible for implementing the function presented at the TSF. Note that this may be the initial module, or there may be several modules, depending on how much pre-conditioning of the inputs is done. It should be noted that one indicator of a pre-conditioning module is that it is invoked for a large number of the TSFI, where the TSFI are all of similar type (e.g., system call). The final aspect of accuracy is that the

mapping makes sense. For instance, mapping a TSFI dealing with access control to a module that checks passwords is not accurate. The evaluator should again use judgement in making this determination. The goal is that this information aids the evaluator in understanding the system and implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the TSF. The bulk of the assessment of whether the SFRs are described accurately by the modules is performed in other work units.

11.8.3.5 Action ADV_TDS.3.2E

ADV_TDS.3-14 The evaluator ***shall examine*** the TOE security functional requirements and the TOE design, to determine that all ST security functional requirements are covered by the TOE design.

831 The evaluator may construct a map between the TOE security functional requirements and the TOE design. This map will likely be from a functional requirement to a set of subsystems. Note that this map may have to be at a level of detail below the subsystem or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

832 For example, the FDP_ACC.1 Subset access control subsystem contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 Subset access control assignment, and these ten rules were implemented in specific places within fifteen modules, it would be inadequate for the evaluator to map FDP_ACC.1 Subset access control to one subsystem and claim the work unit had been completed. Instead, the evaluator would map FDP_ACC.1 Subset access control (rule 1) to subsystem A, behaviours x, y, and z; FDP_ACC.1 Subset access control (rule 2) to subsystem A, behaviours x, p, and q; etc.

ADV_TDS.3-15 The evaluator ***shall examine*** the TOE design to determine that it is an accurate instantiation of all security functional requirements.

833 The evaluator ensures that each security requirement listed in the TOE security functional requirements section of the ST has a corresponding design description in the TOE design that accurately details how the TSF meets that requirement. This requires that the evaluator identify a collection of subsystems that are responsible for implementing a given functional requirement, and then examine those subsystems to understand how the requirement is implemented. Finally, the evaluator would assess whether the requirement was accurately implemented.

834 As an example, if the ST requirements specified a role-based access control mechanism, the evaluator would first identify the subsystems that contribute to this mechanism's implementation. This could be done by in-depth knowledge or understanding of the TOE design or by work done in the previous work unit. Note that this trace is only to identify the subsystems, and is not the complete analysis.

835 The next step would be to understand what mechanism the subsystems implemented. For instance, if the design described an implementation of access control based on UNIX-style protection bits, the design would not be an accurate instantiation of those access control requirements present in the ST example used above. If the evaluator could not determine that the mechanism was accurately implemented because of a lack of detail, the evaluator would have to assess whether all of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for those subsystems.

11.8.4 Evaluation of sub-activity (ADV_TDS.4)

11.8.4.1 Objectives

836 The objective of this sub-activity is to determine whether the TOE design provides a description of the TOE in terms of subsystems sufficient to determine the TSF boundary, and provides a description of the TSF internals in terms of modules (and optionally higher-level abstractions). It provides a detailed description of the SFR-enforcing and SFR-supporting modules and enough information about the SFR-non-interfering modules for the evaluator to determine that the SFRs are completely and accurately implemented; as such, the TOE design provides an explanation of the implementation representation.

11.8.4.2 Input

837 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) security architecture description;

d) the TOE design.

11.8.4.3 Application notes

838 There are three types of activity that the evaluator must undertake with respect to the TOE design. First, the evaluator determines that the TSF boundary has been adequately described. Second, the evaluator determines that the developer has provided documentation that conforms to the content and presentation requirements this subsystem, and that is consistent with other documentation provided for the TOE. Finally, the evaluator must analyse the design information provided for the SFR-enforcing modules (at a detailed level) and the non-SFR-enforcing modules (at a less detailed level) to understand how the system is implemented, and with that knowledge ensure that the TSFI in the functional specification are adequately described, and that the test information adequately tests the TSF (done in the Class ATE: Tests work units).

11.8.4.4 Action ADV_TDS.4.1E

ADV_TDS.4.1C ***The design shall describe the structure of the TOE in terms of subsystems.***

ADV_TDS.4-1 The evaluator ***shall examine*** the TOE design to determine that the structure of the entire TOE is described in terms of subsystems.

839 The evaluator ensures that all of the subsystems of the TOE are identified. This description of the TOE will be used as input to work unit ADV_TDS.4-2, where the parts of the TOE that make up the TSF are identified. That is, this requirement is on the entire TOE rather than on only the TSF.

840 The TOE (and TSF) may be described in multiple layers of abstraction (i.e. subsystems and modules) Depending upon the complexity of the TOE, its design may be described in terms of subsystems and modules, as described in CC Part 3 Annex A.4, ADV_TDS: Subsystems and Modules. For a very simple TOE that can be described solely at the "module" level (see ADV_TDS.4-2), this work unit is not applicable and therefore considered to be satisfied.

841 In performing this activity, the evaluator examines other evidence presented for the TOE (e.g., ST, operator user guidance) to determine that the description of the TOE in such evidence is consistent with the description contained in the TOE design.

ADV_TDS.4.2C ***The design shall describe the TSF in terms of modules, designating each module as SFR-enforcing, SFR-supporting, or SFR-non-interfering.***

ADV_TDS.4-2 The evaluator ***shall examine*** the TOE design to determine that the entire TSF is described in terms of modules.

842 The evaluator will examine the modules for specific properties in other work units; in this work unit the evaluator determines that the modular description covers the entire TSF, and not just a portion of the TSF. The evaluator uses other evidence provided for the evaluation (e.g., functional specification, architectural description) in making this determination. For example, if the functional specification contains interfaces to functionality that does not appear to be described in the TOE design description, it may be the case that a portion of the TSF has not been included appropriately. Making this determination will likely be an iterative process, where as more analysis is done on the other evidence, more confidence can be gained with respect to the completeness of the documentation.

843 Unlike subsystems, modules describe the implementation in a level of detail that can serve as a guide to reviewing the implementation representation. A description of a module should be such that one could create an implementation of the module from the description, and the resulting implementation would be 1) identical to the actual TSF implementation in terms of the interfaces presented and used by the module, and 2) algorithmically identical to the TSF module. For instance, RFC 793 provides a high-level description of the TCP protocol. It is necessarily implementation

independent. While it provides a wealth of detail, it is ***not*** a suitable design description because it is not specific to an implementation. An actual implementation can add to the protocol specified in the RFC, and implementation choices (for instance, the use of global data vs. local data in various parts of the implementation) may have an impact on the analysis that is performed. The design description of the TCP module would list the interfaces presented by the implementation (rather than just those defined in RFC 793), as well as an algorithm description of the processing associated with the modules implementing TCP (assuming it was part of the TSF).

ADV_TDS.4-3 The evaluator ***shall check*** the TOE design to determine that the TSF modules are identified as either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

844 The purpose of designating each module (according to the role a particular module plays in the enforcement of the SFRs) is to allow developers to provide less information about the parts of the TSF that have little role in security. It is always permissible for the developer to provide more information or detail than the requirements demand, as might occur when the information has been gathered outside the evaluation context). In such cases the developer must still designate the modules as either SFR-enforcing, SFR-supporting, or SFR-non-interfering.

845 The accuracy of these designations is continuously reviewed as the evaluation progresses. The concern is the mis-designation of modules as being less important (and hence, having less information) than is really the case. While blatant mis-designations may be immediately apparent (e.g., designating an authentication module as anything but SFR-enforcing when User identification (FIA_UID) is one of the SFRs being claimed), other mis-designations might not be discovered until the TSF is better understood. The evaluator must therefore keep in mind that these designations are the developer's initial best effort, but are subject to change. Further guidance is provided under work unit ADV_TDS.4-16, which examines the accuracy of these designations.

ADV_TDS.4.3C ***The design shall identify all subsystems of the TSF.***

ADV_TDS.4-4 The evaluator ***shall examine*** the TOE design to determine that all subsystems of the TSF are identified.

846 If the design is presented solely in terms of modules, then subsystems in these requirements are equivalent to modules and the activity should be performed at the module level.

847 In work unit ADV_TDS.4-1 all of the subsystems of the TOE were identified, and a determination made that the non-TSF subsystems were correctly characterised. Building on that work, the subsystems that were not characterised as non-TSF subsystems should be precisely identified. The evaluator determines that, of the hardware and software installed and configured according to the Preparative procedures (AGD_PRE) guidance,

each subsystem has been accounted for as either one that is part of the TSF, or one that is not.

ADV_TDS.4.4C ***The design shall provide a description of each subsystem of the TSF.***

ADV_TDS.4-5 The evaluator ***shall examine*** the TOE design to determine that each subsystem of the TSF describes its role in the enforcement of SFRs described in the ST.

848 If the design is presented solely in terms of modules, then this work unit will be considered satisfied by the assessment done in subsequent work units; no explicit action on the part of the evaluator is necessary in this case.

849 On systems that are complex enough to warrant a subsystem-level description of the TSF in addition to the modular description, the goal of the subsystem-level description is to give the evaluator context for the modular description that follows. Therefore, the evaluator ensures that the subsystem-level description contains a description of how the security functional requirements are achieved in the design, but at a level of abstraction above the modular description. This description should discuss the mechanisms used at a level that is aligned with the module description; this will provide the evaluators the road map needed to intelligently assess the information contained in the module description. A well-written set of subsystem descriptions will help guide the evaluator in determining the modules that are most important to examine, thus focusing the evaluation activity on the portions of the TSF that have the most relevance with respect to the enforcement of the SFRs.

850 The evaluator ensures that all subsystems of the TSF have a description. While the description should focus on the role that the subsystem plays in enforcing or supporting the implementation of the SFRs, enough information must be present so that a context for understanding the SFR-related functionality is provided.

ADV_TDS.4.5C ***The design shall provide a description of the interactions among all subsystems of the TSF.***

ADV_TDS.4-6 The evaluator ***shall examine*** the TOE design to determine that interactions between the subsystems of the TSF are described.

851 If the design is presented solely in terms of modules, then this work unit will be considered satisfied by the assessment done in subsequent work units; no explicit action on the part of the evaluator is necessary in this case.

852 On systems that are complex enough to warrant a subsystem-level description of the TSF in addition to the modular description, the goal of describing the interactions between the subsystems is to help provide the reader a better understanding of how the TSF performs it functions. These interactions do not need to be characterised at the implementation level (e.g., parameters passed from one routine in a subsystem to a routine in a different subsystem; global variables; hardware signals (e.g., interrupts) from a

hardware subsystem to an interrupt-handling subsystem), but the data elements identified for a particular subsystem that are going to be used by another subsystem should be covered in this discussion. Any control relationships between subsystems (e.g., a subsystem responsible for configuring a rule base for a firewall system and the subsystem that actually implements these rules) should also be described.

853 It should be noted while the developer should characterise all interactions between subsystems, the evaluators should use their own judgement in assessing the completeness of the description. If the reason for an interaction is unclear, or if there are SFR-related interactions (discovered, for instance, in examining the module-level documentation) that do not appear to be described, the evaluator ensures that this information is provided by the developer. However, if the evaluator can determine that interactions among a particular set of subsystems, while incompletely described by the developer, and a complete description will not aid in understanding the overall functionality nor security functionality provided by the TSF, then the evaluator may choose to consider the description sufficient, and not pursue completeness for its own sake.

ADV_TDS.4.6C ***The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.***

ADV_TDS.4-7 The evaluator ***shall examine*** the TOE design to determine that the mapping between the subsystems of the TSF to the modules of the TSF is complete.

854 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

855 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. To determine completeness, the evaluator examines each mapping and determines that all subsystems map to at least one module, and that all modules map to exactly one subsystem.

ADV_TDS.4-8 The evaluator ***shall examine*** the TOE design to determine that the mapping between the subsystems of the TSF to the modules of the TSF is accurate.

856 If the design is presented solely in terms of modules, then this work unit is considered satisfied.

857 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 mis-understandings 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.4.7C ***The design shall describe each SFR-enforcing and SFR-supporting module in terms of its purpose.***

ADV_TDS.4-9 The evaluator ***shall examine*** the TOE design to determine that the description of the purpose of each SFR-enforcing and SFR-supporting module is complete and accurate.

858 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.

859 The purpose of a module provides a description indicating what function the module is fulfilling. A word of caution to 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).

ADV_TDS.4.8C ***The design shall describe each SFR-enforcing and SFR-supporting module in terms of its SFR-related interfaces, return values from those interfaces, and called interfaces to other modules.***

ADV_TDS.4-10 The evaluator ***shall examine*** the TOE design to determine that the description of the interfaces presented by each SFR-enforcing and SFR-supporting module contain an accurate and complete description of the SFR-related parameters, the invocation conventions for each interface, and any values returned directly by the interface.

860 The SFR-related interfaces of a module are those interfaces used by other modules as a means to invoke the SFR-related operations provided, and to provide inputs to or receive outputs from the module. The purpose in the specification of these interfaces is to permit the exercise of them during

testing. Inter-module interfaces that are not SFR-related need not be specified or described, since they are not a factor in testing. Likewise, other internal interfaces that are not a factor in traversing SFR-related paths of execution (such as those internal paths that are fixed).

861 SFR-related interfaces are described in terms of how they are invoked, and any values that are returned. This description would include a list of parameters, and descriptions of these parameters. Note that global data would also be considered parameters if used by the module (either as inputs or outputs) when invoked. If a parameter were expected to take on a set of values (e.g., a "flag" parameter), the complete set of values the parameter could take on that would have an effect on module processing would be specified. Likewise, parameters representing data structures are described such that each field of the data structure is identified and described. Note that different programming languages may have additional "interfaces" that would be non-obvious; an example would be operator/function overloading in C++. This "implicit interface" in the class description would also be described as part of the low-level TOE design. Note that although a module could present only one interface, it is more common that a module presents a small set of related interfaces.

862 In terms of the assessment of parameters (inputs and outputs) to a module, any use of global data must also be considered. A module "uses" global data if it either reads or writes the data. In order to assure the description of such parameters (if used) is complete, the evaluator uses other information provided about the module in the TOE design (interfaces, algorithmic description, etc.), as well as the description of the particular set of global data assessed in work unit ADV_TDS.4-9. For instance, the evaluator could first determine the processing the module performs by examining its function and interfaces presented (particularly the parameters of the interfaces). They could then check to see if the processing appears to "touch" any of the global data areas identified in the TDS design. The evaluator then determines that, for each global data area that appears to be "touched", that global data area is listed as a means of input or output by the module the evaluator is examining.

863 Invocation conventions are a programming-reference-type description that one could use to correctly invoke a module's interface if one were writing a program to make use of the module's functionality through that interface. This includes necessary inputs and outputs, including any set-up that may need to be performed with respect to global variables.

864 Values returned through the interface refer to values that are either passed through parameters or messages; values that the function call itself returns in the style of a "C" program function call; or values passed through global means (such as certain error routines in *ix-style operating systems).

865 In order to assure the description is complete, the evaluator uses other information provided about the module in the TOE design (e.g., algorithmic description, global data used) to ensure that it appears all data necessary for performing the functions of the module is presented to the module, and that

any values that other modules expect the module under examination to provide are identified as being returned by the module. The evaluator determines accuracy by ensuring that the description of the processing matches the information listed as being passed to or from an interface.

866 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 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.4.10C element, which maps the TSFI in the functional specification to the modules of the TSF.

ADV_TDS.4.9C ***The design shall describe each SFR-non-interfering module in terms of its purpose and interaction with other modules.***

ADV_TDS.4-11 The evaluator ***shall examine*** the TOE design to determine that SFR-non-interfering modules are correctly categorised.

867 As mentioned in work unit ADV_TDS.4-3, less information is required about modules that are SFR-non-interfering. 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-non-interfering and the evaluation (information about other modules in the TOE design, the functional specification, the security architecture description, the administrative guidance, the TSF internals document, and perhaps even the implementation representation) whether the module is indeed 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-non-interfering, even though it is labelled as such. However, if the evidence provided indicates that a SFR-non-interfering module is SFR-enforcing or SFR-supporting, the evaluator requests additional information from the developer in order to resolve the apparent inconsistency. For example, 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, it is discovered that the only information the developer has provided is a purpose and a set of interactions (thus implicitly categorising Module B as non-SFR-enforcing). 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.

868 Another example would be where the evaluator examines the mapping of the TSFI to the modules as provided by ADV_TDS.4.2D. 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-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 or SFR-supporting. Again, the evaluator should approach the developer to resolve this discrepancy.

869 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.

ADV_TDS.4-12 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.

870 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.

871 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 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.

ADV_TDS.4-13 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.

872 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.

873 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 non-SFR-enforcing modules in the overall TOE design. Understanding of this role will aid the evaluator in performing work unit ADV_TDS.4-6.

874 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; the interactions should describe what the module depends on from other modules in order to accomplish this function.

875 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 that the interactions are accurately and completely described.

ADV_TDS.4.10C ***The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.***

ADV_TDS.4-14 The evaluator ***shall examine*** the TOE design to determine that it contains a complete and accurate mapping from the TSFI described in the functional specification to the modules of the TSF described in the TOE design.

876 The modules described in the TOE design provide a description of the implementation of the TSF. The TSFI provide a description of how the implementation is exercised. The evidence from the developer identifies the module that is initially invoked when an operation is requested at the TSFI, and identify the chain of modules invoked up to the module that is primarily responsible for implementing the functionality. However, a complete call tree for each TSFI is not required for this work unit. The cases in which more than one module would have to be identified are where there are “entry point” modules or wrapper modules that have no functionality other than conditioning inputs or de-multiplexing an input. Mapping to one of these modules would not provide any useful information to the evaluator.

877 The evaluator assesses the completeness of the mapping by ensuring that all of the TSFI map to at least one module. The verification of accuracy is more complex.

878 The first aspect of accuracy is that each TSFI is mapped to a module at the TSF boundary. This determination can be made by reviewing the module description and its interfaces/interactions. The next aspect of accuracy is that each TSFI identifies a chain of modules between the initial module identified and a module that is primarily responsible for implementing the function presented at the TSF. Note that this may be the initial module, or there may be several modules, depending on how much pre-conditioning of the inputs is done. It should be noted that one indicator of a pre-conditioning module is that it is invoked for a large number of the TSFI, where the TSFI are all of similar type (e.g., system call). The final aspect of accuracy is that the mapping makes sense. For instance, mapping a TSFI dealing with access control to a module that checks passwords is not accurate. The evaluator should again use judgement in making this determination. The goal is that this information aids the evaluator in understanding the system and implementation of the SFRs, and ways in which entities at the TSF boundary can interact with the TSF. The bulk of the assessment of whether the SFRs are described accurately by the modules is performed in other work units.

11.8.4.5 Action ADV_TDS.4.2E

ADV_TDS.4-15 The evaluator ***shall examine*** the TOE security functional requirements and the TOE design, to determine that all ST security functional requirements are covered by the TOE design.

879 The evaluator may construct a map between the TOE security functional requirements and the TOE design. This map will likely be from a functional requirement to a set of subsystems. Note that this map may have to be at a level of detail below the subsystem or even element level of the requirements, because of operations (assignments, refinements, selections) performed on the functional requirement by the ST author.

880 For example, the FDP_ACC.1 Subset access control subsystem contains an element with assignments. If the ST contained, for instance, ten rules in the FDP_ACC.1 Subset access control assignment, and these ten rules were implemented in specific places within fifteen modules, it would be inadequate for the evaluator to map FDP_ACC.1 Subset access control to one subsystem and claim the work unit had been completed. Instead, the evaluator would map FDP_ACC.1 Subset access control (rule 1) to subsystem A, behaviours x, y, and z; FDP_ACC.1 Subset access control (rule 2) to subsystem A, behaviours x, p, and q; etc.

ADV_TDS.4-16 The evaluator ***shall examine*** the TOE design to determine that it is an accurate instantiation of all security functional requirements.

881 The evaluator ensures that each security requirement listed in the TOE security functional requirements section of the ST has a corresponding design description in the TOE design that accurately details how the TSF

meets that requirement. This requires that the evaluator identify a collection of subsystems that are responsible for implementing a given functional requirement, and then examine those subsystems to understand how the requirement is implemented. Finally, the evaluator would assess whether the requirement was accurately implemented.

882 As an example, if the ST requirements specified a role-based access control mechanism, the evaluator would first identify the subsystems that contribute to this mechanism's implementation. This could be done by in-depth knowledge or understanding of the TOE design or by work done in the previous work unit. Note that this trace is only to identify the subsystems, and is not the complete analysis.

883 The next step would be to understand what mechanism the subsystems implemented. For instance, if the design described an implementation of access control based on UNIX-style protection bits, the design would not be an accurate instantiation of those access control requirements present in the ST example used above. If the evaluator could not determine that the mechanism was accurately implemented because of a lack of detail, the evaluator would have to assess whether all of the SFR-enforcing subsystems have been identified, or if adequate detail had been provided for those subsystems.

11.8.5 Evaluation of sub-activity (ADV_TDS.5)

884 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

11.8.6 Evaluation of sub-activity (ADV_TDS.6)

885 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

12 Class AGD: Guidance documents

12.1 Introduction

886 The purpose of the guidance document activity is to judge the adequacy of the documentation describing how the user can handle the TOE in a secure manner. Such documentation should take into account the various types of users (e.g. those who accept, install, administrate or operate the TOE) whose incorrect actions could adversely affect the security of the TOE or of their own data.

887 The guidance documents class is subdivided into two families which are concerned firstly with the preparative user guidance (all that has to be done to transform the delivered TOE into its evaluated configuration in the environment as described in the ST, i.e. accepting and installing the TOE) and secondly with the operational user guidance (all that has to be done during the operation of the TOE in its evaluated configuration, i.e. operation and administration).

12.2 Application notes

888 The guidance documents activity applies to those functions and interfaces which are related to the security of the TOE. The secure configuration of the TOE is described in the ST.

12.3 Operational user guidance (AGD_OPE)

12.3.1 Evaluation of sub-activity (AGD_OPE.1)

12.3.1.1 Objectives

889 The objectives of this sub-activity are to determine whether the user guidance describes for each user role the security functionality and interfaces provided by the TSF, provides instructions and guidelines for the secure use of the TOE, addresses secure procedures for all modes of operation, facilitates prevention and detection of insecure TOE states, or whether it is misleading or unreasonable.

12.3.1.2 Input

890 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design, if applicable;

d) the user guidance;

12.3.1.3 Action AGD_OPE.1.1E

AGD_OPE.1.1C ***The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.***

AGD_OPE.1-1 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

891 The configuration of the TOE may allow different user roles to have dissimilar privileges in making use of the different functions of the TOE. This means that some users are authorised to perform certain functions, while other users may not be so authorised. These functions and privileges should be described, for each user role, by the user guidance.

892 The user guidance identifies, for each user role, the functions and privileges that must be controlled, the types of commands required for them, and the reasons for such commands. The user guidance should contain warnings regarding the use of these functions and privileges. Warnings should address expected effects, possible side effects, and possible interactions with other functions and privileges.

AGD_OPE.1.2C ***The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.***

AGD_OPE.1-2 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each user role, the secure use of the available interfaces provided by the TOE.

893 The user guidance should provide advice regarding effective use of the TSF (e.g. reviewing password composition practises, suggested frequency of user file backups, discussion on the effects of changing user access privileges).

AGD_OPE.1.3C ***The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.***

AGD_OPE.1-3 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each user role, the available security functionality and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

894 The user guidance should contain an overview of the security functionality that is visible at the user interfaces.

895 The user guidance should identify and describe the purpose, behaviour, and interrelationships of the security interfaces and functionality.

896 For each user-accessible interface, the user guidance should:

a) describe the method(s) by which the interface is invoked (e.g. command-line, programming-language system call, menu selection, command button);

b) describe the parameters to be set by the user, their particular purposes, valid and default values, and secure and insecure use settings of such parameters, both individually or in combination;

c) describe the immediate TSF response, message, or code returned.

897 The evaluator should consider the functional specification and the ST to determine that the TSF described in these documents is consistent to the operational user guidance. The evaluator has to ensure that the operational user guidance is complete to allow the secure use through the TSFI available to all types of human users. The evaluator may, as an aid, prepare an informal mapping between the guidance and these documents. Any omissions in this mapping may indicate incompleteness.

AGD_OPE.1.4C ***The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.***

AGD_OPE.1-4 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each user role, each type of security-relevant event relative to the user functions that need to be performed, including changing the

security characteristics of entities under the control of the TSF and operation following failure or operational error.

898 All types of security-relevant events are detailed for each user role, such that each user knows what events may occur and what action (if any) he may have to take in order to maintain security. Security-relevant events that may occur during operation of the TOE (e.g. audit trail overflow, system crash, updates to user records, such as when a user account is removed when the user leaves the organisation) are adequately defined to allow user intervention to maintain secure operation.

AGD_OPE.1.5C ***The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.***

AGD_OPE.1-5 The evaluator ***shall examine*** the operational user guidance and other evaluation evidence to determine that the guidance identifies all possible modes of operation of the TOE (including, if applicable, operation following failure or operational error), their consequences and implications for maintaining secure operation.

899 Other evaluation evidence, particularly the functional specification, provide an information source that the evaluator should use to determine that the guidance contains sufficient guidance information.

900 If test documentation is included in the assurance package, then the information provided in this evidence can also be used to determine that the guidance contains sufficient guidance documentation. The detail provided in the test steps can be used to confirm that the guidance provided is sufficient for the use and administration of the TOE.

901 The evaluator should focus on a single human visible TSFI at a time, comparing the guidance for securely using the TSFI with other evaluation evidence, to determine that the guidance related to the TSFI is sufficient for the secure usage (i.e. consistent with the SFRs) of that TSFI. The evaluator should also consider the relationships between interfaces, searching for potential conflicts.

AGD_OPE.1.6C ***The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.***

AGD_OPE.1-6 The evaluator ***shall examine*** the operational user guidance to determine that it describes, for each user role, the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

902 The evaluator analyses the security objectives for the operational environment in the ST and determines that for each user role, the relevant security measures are described appropriately in the user guidance.

903 The security measures described in the user guidance should include all relevant external procedural, physical, personnel and connectivity measures.

904 Note that those measures relevant for secure installation of the TOE are examined in Preparative procedures (AGD_PRE).

AGD_OPE.1.7C ***The operational user guidance shall be clear and reasonable.***

AGD_OPE.1-7 The evaluator ***shall examine*** the operational user guidance to determine that it is clear.

905 The guidance is unclear if it can reasonably be misconstrued by an administrator or user, and used in a way detrimental to the TOE, or to the security provided by the TOE.

AGD_OPE.1-8 The evaluator ***shall examine*** the operational user guidance to determine that it is reasonable.

906 The guidance is unreasonable if it makes demands on the TOE's usage or operational environment that are inconsistent with the ST or unduly onerous to maintain security.

12.4 Preparative procedures (AGD_PRE)

12.4.1 Evaluation of sub-activity (AGD_PRE.1)

12.4.1.1 Objectives

907 The objective of this sub-activity is to determine whether the procedures and steps for the secure preparation of the TOE have been documented and result in a secure configuration.

12.4.1.2 Input

908 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE including its preparative procedures;

c) the description of developer's delivery procedures, if applicable;

12.4.1.3 Application notes

909 The preparative procedures refer to all acceptance and installation procedures, that are necessary to progress the TOE to the secure configuration as described in the ST.

12.4.1.4 Action AGD_PRE.1.1E

AGD_PRE.1.1C ***The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.***

AGD_PRE.1-1 The evaluator ***shall check*** that the procedures necessary for the secure acceptance of the delivered TOE have been provided.

910 If it is not anticipated by the developer's delivery procedures that acceptance procedures will or can be applied, this work unit is not applicable, and is therefore considered to be satisfied.

AGD_PRE.1-2 The evaluator ***shall examine*** the provided acceptance procedures to determine that they describe the steps necessary for secure acceptance of the TOE in accordance with the developer's delivery procedures.

911 The acceptance procedures should include as a minimum, that the user has to check that all parts of the TOE as indicated in the ST have been delivered in the correct version.

912 The acceptance procedures should reflect the steps the user has to perform in order to accept the delivered TOE that are implied by the developer's delivery procedures.

913 The acceptance procedures should provide detailed information about the following, if applicable:

a) making sure that the delivered TOE is the complete evaluated instance;

b) detecting modification/masquerading of the delivered TOE.

AGD_PRE.1.2C ***The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.***

AGD_PRE.1-3 The evaluator ***shall check*** that the procedures necessary for the secure installation of the TOE have been provided.

914 If it is not anticipated that installation procedures will or can be applied for the TOE and the operational environment (e.g. because the TOE may already be delivered in an operational state and there are no requirements for the environment) this work unit is not applicable, and is therefore considered to be satisfied.

AGD_PRE.1-4 The evaluator ***shall examine*** the provided installation procedures to determine that they describe the steps necessary for secure installation of the TOE and the secure preparation of the operational environment in accordance with the security objectives in the ST.

915 If it is not anticipated that installation procedures will or can be applied (e.g. because the TOE may already be delivered in an operational state), this work unit is not applicable, and is therefore considered to be satisfied.

916 The installation procedures should provide detailed information about the following, if applicable:

a) minimum system requirements for secure installation;

b) requirements for the operational environment in accordance with the security objectives provided by the ST;

c) changing the installation specific security characteristics of entities under the control of the TSF (for example parameters, settings, passwords);

d) handling exceptions and problems.

12.4.1.5 Action AGD_PRE.1.2E

AGD_PRE.1-5 The evaluator ***shall perform*** all user procedures necessary to prepare the TOE to determine that the TOE and its operational environment can be prepared securely using only the supplied preparative user guidance.

917 Preparation requires the evaluator to advance the TOE from a deliverable state to the state in which it is operational, including acceptance and installation of the TOE, and enforcing the SFRs consistent with the security objectives for the TOE specified in the ST.

918 The evaluator should follow only the developer's procedures and may perform the activities that customers are usually expected to perform to accept and install the TOE, using the supplied preparative guidance documentation only. Any difficulties encountered during such an exercise may be indicative of incomplete, unclear or unreasonable guidance.

919 This work unit may be performed in conjunction with the evaluation activities under Independent testing (ATE_IND).

920 If it is known that the TOE will be used as a dependent component for a composed TOE evaluation, then the evaluator should ensure that the operational environment is satisfied by the base component used in the composed TOE.

13 Class ALC: Life-cycle support

13.1 Introduction

921 The purpose of the life-cycle support activity is to determine the adequacy of the security procedures that the developer uses during the development and maintenance of the TOE. These procedures include the life-cycle model used by the developer, the configuration management, the security measures used throughout TOE development, the tools used by the developer throughout the life-cycle of the TOE, the handling of security flaws, and the delivery activity.

922 Poorly controlled development and maintenance of the TOE can result in vulnerabilities in the implementation. Conformance to a defined life-cycle model can help to improve controls in this area. A measurable life-cycle model used for the TOE can remove ambiguity in assessing the development progress of the TOE.

923 The purpose of the configuration management activity is to assist the consumer in identifying the evaluated TOE, to ensure that configuration items are uniquely identified, and the adequacy of the procedures that are used by the developer to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.

924 Developer security procedures are intended to protect the TOE and its associated design information from interference or disclosure. Interference in the development process may allow the deliberate introduction of vulnerabilities. Disclosure of design information may allow vulnerabilities to be more easily exploited. The adequacy of the procedures will depend on the nature of the TOE and the development process.

925 The use of well-defined development tools and the application of implementation standards by the developer and by third parties involved in the development process help to ensure that vulnerabilities are not inadvertently introduced during refinement.

926 The flaw remediation activity is intended to track security flaws, to identify corrective actions, and to distribute the corrective action information to TOE users.

927 The purpose of the delivery activity is to judge the adequacy of the documentation of the procedures used to ensure that the TOE is delivered to the consumer without modification.

13.2 CM capabilities (ALC_CMC)

13.2.1 Evaluation of sub-activity (ALC_CMC.1)

13.2.1.1 Objectives

928 The objectives of this sub-activity are to determine whether the developer has clearly identified the TOE.

13.2.1.2 Input

929 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing.

13.2.1.3 Action ALC_CMC.1.1E

ALC_CMC.1.1C ***The TOE shall be labelled with its unique reference.***

ALC_CMC.1-1 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

930 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

931 The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

932 Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMC.1-2 The evaluator ***shall check*** that the TOE references used are consistent.

933 If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

934 The evaluator also verifies that the TOE reference is consistent with the ST.

935 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

13.2.2 Evaluation of sub-activity (ALC_CMC.2)

13.2.2.1 Objectives

936 The objectives of this sub-activity are to determine whether the developer uses a CM system that uniquely identifies all configuration items.

13.2.2.2 Input

937 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing;

c) the configuration management documentation.

13.2.2.3 Application notes

938 This component contains an implicit evaluator action to determine that the CM system is being used. As the requirements here are limited to identification of the TOE and provision of a configuration list, this action is already covered by, and limited to, the existing work units. At Evaluation of sub-activity (ALC_CMC.3) the requirements are expanded beyond these two items, and more explicit evidence of operation is required.

13.2.2.4 Action ALC_CMC.2.1E

ALC_CMC.2.1C ***The TOE shall be labelled with its unique reference.***

ALC_CMC.2-1 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

939 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

940 The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

941 Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMC.2-2 The evaluator ***shall check*** that the TOE references used are consistent.

942 If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

943 The evaluator also verifies that the TOE reference is consistent with the ST.

944 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMC.2.2C ***The CM documentation shall describe the method used to uniquely identify the configuration items.***

ALC_CMC.2-3 The evaluator ***shall examine*** the method of identifying configuration items to determine that it describes how configuration items are uniquely identified.

945 Procedures should describe how the status of each configuration item can be tracked throughout the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item.

ALC_CMC.2.3C ***The CM system shall uniquely identify all configuration items.***

ALC_CMC.2-4 The evaluator ***shall examine*** the configuration items to determine that they are identified in a way that is consistent with the CM documentation.

946 Assurance that the CM system uniquely identifies all configuration items is gained by examining the identifiers for the configuration items. For both configuration items that comprise the TOE, and drafts of configuration items that are submitted by the developer as evaluation evidence, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

13.2.3 Evaluation of sub-activity (ALC_CMC.3)

13.2.3.1 Objectives

947 The objectives of this sub-activity are to determine whether the developer uses a CM system that uniquely identifies all configuration items, and whether the ability to modify these items is properly controlled.

13.2.3.2 Input

948 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing;

c) the configuration management documentation.

13.2.3.3 Action ALC_CMC.3.1E

ALC_CMC.3.1C ***The TOE shall be labelled with its unique reference.***

ALC_CMC.3-1 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

949 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

950 The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

951 Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMC.3-2 The evaluator ***shall check*** that the TOE references used are consistent.

952 If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

953 The evaluator also verifies that the TOE reference is consistent with the ST.

954 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMC.3.2C ***The CM documentation shall describe the method used to uniquely identify the configuration items.***

ALC_CMC.3-3 The evaluator ***shall examine*** the method of identifying configuration items to determine that it describes how configuration items are uniquely identified.

955 Procedures should describe how the status of each configuration item can be tracked throughout the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item.

ALC_CMC.3.3C ***The CM system shall uniquely identify all configuration items.***

ALC_CMC.3-4 The evaluator ***shall examine*** the configuration items to determine that they are identified in a way that is consistent with the CM documentation.

956 Assurance that the CM system uniquely identifies all configuration items is gained by examining the identifiers for the configuration items. For both configuration items that comprise the TOE, and drafts of configuration items that are submitted by the developer as evaluation evidence, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

ALC_CMC.3.4C ***The CM system shall provide measures such that only authorised changes are made to the configuration items.***

ALC_CMC.3-5 The evaluator ***shall examine*** the CM access control measures described in the CM plan to determine that they are effective in preventing unauthorised access to the configuration items.

957 The evaluator may use a number of methods to determine that the CM access control measures are effective. For example, the evaluator may exercise the access control measures to ensure that the procedures could not be bypassed. The evaluator may use the outputs generated by the CM system procedures required by ALC_CMC.3.8C. The evaluator may also witness a demonstration of the CM system to ensure that the access control measures employed are operating effectively.

ALC_CMC.3.5C ***The CM documentation shall include a CM plan.***

ALC_CMC.3-6 The evaluator ***shall check*** that the CM documentation provided includes a CM plan. The CM plan needs not to be a connected document, but it is recommended that there is a single document that describes where the various parts of the CM plan can be found. If the CM plan is no single document, the list in the following work unit gives hints regarding which context is expected.

ALC_CMC.3.6C ***The CM plan shall describe how the CM system is used for the development of the TOE.***

ALC_CMC.3-7 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used for the development of the TOE.

958 The descriptions contained in a CM plan include, if applicable:

a) all activities performed in the TOE development that are subject to configuration management procedures (e.g. creation, modification or deletion of a configuration item, data-backup, archiving);

b) which means (e.g. CM tools, forms) have to be made available;

c) the usage of the CM tools: the necessary details for a user of the CM system to be able to operate the CM tools correctly in order to maintain the integrity of the TOE;

d) which other objects (development components, tools, assessment environments, etc) are taken under CM control;

e) the roles and responsibilities of individuals required to perform operations on individual configuration items (different roles may be identified for different types of configuration items (e.g. design documentation or source code));

f) how CM instances (e.g. change control boards, interface control working groups) are introduced and staffed;

g) the description of the change management;

h) the procedures that are used to ensure that only authorised individuals can make changes to configuration items;

i) the procedures that are used to ensure that concurrency problems do not occur as a result of simultaneous changes to configuration items;

j) the evidence that is generated as a result of application of the procedures. For example, for a change to a configuration item, the CM system might record a description of the change, accountability for the change, identification of all configuration items affected, status (e.g. pending or completed), and date and time of the change. This might be recorded in an audit trail of changes made or change control records;

k) the approach to version control and unique referencing of TOE versions (e.g. covering the release of patches in operating systems, and the subsequent detection of their application).

ALC_CMC.3.7C ***The evidence shall demonstrate that all configuration items are being maintained under the CM system.***

ALC_CMC.3-8 The evaluator ***shall check*** that the configuration items identified in the configuration list are being maintained by the CM system.

959 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator should check that for each type of configuration item (e.g. design documents or source code modules) contained in the configuration list there are examples of the evidence generated by the procedures described in the CM plan. In this case, the approach to sampling will depend upon the level of granularity used in the CM system to control CM items. Where, for example, 10,000 source code modules are identified in the configuration list, a different sampling strategy should be applied compared to the case in which there are only 5, or even 1. The emphasis of this activity should be on ensuring that the CM system is being operated correctly, rather than on the detection of any minor error.

960 For guidance on sampling see A.2, Sampling.

ALC_CMC.3.8C ***The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.***

ALC_CMC.3-9 The evaluator ***shall check*** the CM documentation to ascertain that it includes the CM system records identified by the CM plan.

961 The output produced by the CM system should provide the evidence that the evaluator needs to be confident that the CM plan is being applied, and also that all configuration items are being maintained by the CM system as required by ALC_CMC.3.7C. Example output could include change control forms, or configuration item access approval forms.

ALC_CMC.3-10 The evaluator ***shall examine*** the evidence to determine that the CM system is being operated in accordance with the CM plan.

962 The evaluator should select and examine a sample of evidence covering each type of CM-relevant operation that has been performed on a configuration item (e.g. creation, modification, deletion, reversion to an earlier version) to confirm that all operations of the CM system have been carried out in line with documented procedures. The evaluator confirms that the evidence includes all the information identified for that operation in the CM plan. Examination of the evidence may require access to a CM tool that is used. The evaluator may choose to sample the evidence.

963 For guidance on sampling see A.2, Sampling.

964 Further confidence in the correct operation of the CM system and the effective maintenance of configuration items may be established by means of interview with selected development staff. In conducting such interviews, the evaluator should aim to gain a deeper understanding of how the CM system is used in practise as well as to confirm that the CM procedures are being applied as described in the CM documentation. Note that such interviews should complement rather than replace the examination of documentary evidence, and may not be necessary if the documentary evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and records alone. This is one case where clarification may be necessary through interviews.

965 It is expected that the evaluator will visit the development site in support of this activity.

966 For guidance on site visits see A.4, Site Visits.

13.2.4 Evaluation of sub-activity (ALC_CMC.4)

13.2.4.1 Objectives

967 The objectives of this sub-activity are to determine whether the developer has clearly identified the TOE and its associated configuration items, and whether the ability to modify these items is properly controlled by automated

tools, thus making the CM system less susceptible to human error or negligence.

13.2.4.2 Input

968 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing;

c) the configuration management documentation.

13.2.4.3 Action ALC_CMC.4.1E

ALC_CMC.4.1C ***The TOE shall be labelled with its unique reference.***

ALC_CMC.4-1 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

969 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

970 The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

971 Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMC.4-2 The evaluator ***shall check*** that the TOE references used are consistent.

972 If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

973 The evaluator also verifies that the TOE reference is consistent with the ST.

974 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be

labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMC.4.2C ***The CM documentation shall describe the method used to uniquely identify the configuration items.***

ALC_CMC.4-3 The evaluator ***shall examine*** the method of identifying configuration items to determine that it describes how configuration items are uniquely identified.

975 Procedures should describe how the status of each configuration item can be tracked throughout the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item.

ALC_CMC.4.3C ***The CM system shall uniquely identify all configuration items.***

ALC_CMC.4-4 The evaluator ***shall examine*** the configuration items to determine that they are identified in a way that is consistent with the CM documentation.

976 Assurance that the CM system uniquely identifies all configuration items is gained by examining the identifiers for the configuration items. For both configuration items that comprise the TOE, and drafts of configuration items that are submitted by the developer as evaluation evidence, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

ALC_CMC.4.4C ***The CM system shall provide automated measures such that only authorised changes are made to the configuration items.***

ALC_CMC.4-5 The evaluator ***shall examine*** the CM access control measures described in the CM plan (cf. ALC_CMC.4.6C) to determine that they are automated and effective in preventing unauthorised access to the configuration items.

977 The evaluator may use a number of methods to determine that the CM access control measures are effective. For example, the evaluator may exercise the access control measures to ensure that the procedures could not be bypassed.

The evaluator may use the outputs generated by the CM system procedures required by ALC_CMC.4.10C. The evaluator may also witness a demonstration of the CM system to ensure that the access control measures employed are operating effectively.

ALC_CMC.4.5C ***The CM system shall support the production of the TOE by automated means.***

ALC_CMC.4-6 The evaluator ***shall check*** the CM plan (cf. ALC_CMC.4.6C) for automated procedures for supporting the production of the TOE.

978 The term “production” applies to those processes adopted by the developer to progress the TOE from the implementation representation to a state acceptable for delivery to the end customer.

979 The evaluator verifies the existence of automated production support procedures within the CM plan.

980 The following are examples for automated means supporting the production of the TOE:

– a “make” tool (as provided with many software development tools) in the case of a software TOE;

– a tool ensuring automatically (for example by means of bar codes) that only parts are combined which indeed belong together in the case of a hardware TOE.

ALC_CMC.4-7 The evaluator ***shall examine*** the TOE production support procedures to determine that they are effective in ensuring that a TOE is generated that reflects its implementation representation.

981 The production support procedures should describe which tools have to be used to produce the final TOE from the implementation representation in a clearly defined way. The conventions, directives, or other necessary constructs are described under ALC_TAT.

982 The evaluator determines that by following the production support procedures the correct configuration items would be used to generate the TOE. For example, in a software TOE this may include checking that the automated production procedures ensure that all source files and related libraries are included in the compiled object code. Moreover, the procedures should ensure that compiler options and comparable other options are defined uniquely. For a hardware TOE, this work unit may include checking that the automatic production procedures ensure that the belonging parts are built together and no parts are missing.

983 The customer can then be confident that the version of the TOE delivered for installation is derived from the implementation representation in an unambiguous way and implements the SFRs as described in the ST.

984 The evaluator should bear in mind that the CM system need not necessarily possess the capability to produce the TOE, but should provide support for the process that will help reduce the probability of human error.

ALC_CMC.4.6C ***The CM documentation shall include a CM plan.***

ALC_CMC.4-8 The evaluator ***shall check*** that the CM documentation provided includes a CM plan. The CM plan needs not to be a connected document, but it is recommended that there is a single document that describes where the various parts of the CM plan can be found. If the CM plan is no single document, the list in the following work unit gives hints regarding which context is expected.

ALC_CMC.4.7C ***The CM plan shall describe how the CM system is used for the development of the TOE.***

ALC_CMC.4-9 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used for the development of the TOE.

985 The descriptions contained in a CM plan include, if applicable:

a) all activities performed in the TOE development that are subject to configuration management procedures (e.g. creation, modification or deletion of a configuration item, data-backup, archiving);

b) which means (e.g. CM tools, forms) have to be made available;

c) the usage of the CM tools: the necessary details for a user of the CM system to be able to operate the CM tools correctly in order to maintain the integrity of the TOE;

d) the production support procedures;

e) which other objects (development components, tools, assessment environments, etc) are taken under CM control;

f) the roles and responsibilities of individuals required to perform operations on individual configuration items (different roles may be identified for different types of configuration items (e.g. design documentation or source code));

g) how CM instances (e.g. change control boards, interface control working groups) are introduced and staffed;

h) the description of the change management;

i) the procedures that are used to ensure that only authorised individuals can make changes to configuration items;

j) the procedures that are used to ensure that concurrency problems do not occur as a result of simultaneous changes to configuration items;

k) the evidence that is generated as a result of application of the procedures. For example, for a change to a configuration item, the CM system might record a description of the change, accountability for the change, identification of all configuration items affected, status (e.g. pending or completed), and date and time of the change. This might be recorded in an audit trail of changes made or change control records;

l) the approach to version control and unique referencing of TOE versions (e.g. covering the release of patches in operating systems, and the subsequent detection of their application).

ALC_CMC.4.8C ***The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.***

ALC_CMC.4-10 The evaluator ***shall examine*** the CM plan to determine that it describes the procedures used to accept modified or newly created configuration items as parts of the TOE.

986 The descriptions of the acceptance procedures in the CM plan should include the developer roles or individuals responsible for the acceptance and the criteria to be used for acceptance. They should take into account all acceptance situations that may occur, in particular:

a) accepting an item into the CM system for the first time, in particular inclusion of software, firmware and hardware components from other manufacturers into the TOE (“integration”);

b) moving configuration items to the next life-cycle phase at each stage of the construction of the TOE (e.g. module, subsystem, system);

c) subsequent to transports between different development sites.

987 If this work unit is applied to a dependent component that is going to be integrated in a composed TOE, the CM plan should consider the control of base components obtained by the dependent TOE developer.

988 When obtaining the components the evaluators are to verify the following:

a) Transfer of each base component from the base component developer to the integrator (dependent TOE developer) was performed in accordance with the base component TOE's secure delivery procedures, as reported in the base component TOE certification report.

b) The component received has the same identifiers as those stated in the ST and Certification Report for the component TOE.

c) All additional material required by a developer for composition (integration) is provided. This is to include the necessary extract of the component TOE's functional specification.

ALC_CMC.4.9C ***The evidence shall demonstrate that all configuration items are being maintained under the CM system.***

ALC_CMC.4-11 The evaluator ***shall check*** that the configuration items identified in the configuration list are being maintained by the CM system.

989 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator should check that for each type of configuration item (e.g. design documents or source code modules) contained in the configuration list there are examples of the evidence generated by the procedures described in the CM plan. In this case, the approach to sampling will depend upon the level of granularity used in the CM system to control CM items. Where, for example, 10,000 source code modules are identified in the configuration list, a different sampling strategy should be applied compared to the case in which there are only 5, or even 1. The emphasis of this activity should be on ensuring that the CM system is being operated correctly, rather than on the detection of any minor error.

990 For guidance on sampling see A.2, Sampling.

ALC_CMC.4.10C ***The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.***

ALC_CMC.4-12 The evaluator ***shall check*** the CM documentation to ascertain that it includes the CM system records identified by the CM plan.

991 The output produced by the CM system should provide the evidence that the evaluator needs to be confident that the CM plan is being applied, and also that all configuration items are being maintained by the CM system as required by ALC_CMC.4.9C. Example output could include change control forms, or configuration item access approval forms.

ALC_CMC.4-13 The evaluator ***shall examine*** the evidence to determine that the CM system is being operated in accordance with the CM plan.

992 The evaluator should select and examine a sample of evidence covering each type of CM-relevant operation that has been performed on a configuration item (e.g. creation, modification, deletion, reversion to an earlier version) to confirm that all operations of the CM system have been carried out in line with documented procedures. The evaluator confirms that the evidence includes all the information identified for that operation in the CM plan. Examination of the evidence may require access to a CM tool that is used. The evaluator may choose to sample the evidence.

993 For guidance on sampling see A.2, Sampling.

994 Further confidence in the correct operation of the CM system and the effective maintenance of configuration items may be established by means of interviews with selected development staff. In conducting such interviews, the evaluator should aim to gain a deeper understanding of how the CM system is used in practise as well as to confirm that the CM procedures are

being applied as described in the CM documentation. Note that such interviews should complement rather than replace the examination of documentary evidence, and may not be necessary if the documentary evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and records alone. This is one case where clarification may be necessary through interviews.

995 It is expected that the evaluator will visit the development site in support of this activity.

996 For guidance on site visits see A.4, Site Visits.

13.2.5 Evaluation of sub-activity (ALC_CMC.5)

13.2.5.1 Objectives

997 The objectives of this sub-activity are to determine whether the developer has clearly identified the TOE and its associated configuration items, and whether the ability to modify these items is properly controlled by automated tools, thus making the CM system less susceptible to human error or negligence.

13.2.5.2 Input

998 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing;

c) the configuration management documentation.

13.2.5.3 Action ALC_CMC.5.1E

ALC_CMC.5.1C ***The TOE shall be labelled with its unique reference.***

ALC_CMC.5-1 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

999 The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

1000 The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

1001 Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMC.5-2 The evaluator ***shall check*** that the TOE references used are consistent.

1002 If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

1003 The evaluator also verifies that the TOE reference is consistent with the ST.

1004 If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMC.5.2C ***The CM documentation shall describe the method used to uniquely identify the configuration items.***

ALC_CMC.5-3 The evaluator ***shall examine*** the method of identifying configuration items to determine that it describes how configuration items are uniquely identified.

1005 Procedures should describe how the status of each configuration item can be tracked throughout the life-cycle of the TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item.

ALC_CMC.5.3C ***The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.***

ALC_CMC.5-4 The evaluator ***shall examine*** the CM documentation to determine that it justifies that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.

1006 The CM documentation should make it sufficiently clear that by following the acceptance procedures only parts of adequate quality are incorporated into the TOE.

ALC_CMC.5.4C ***The CM system shall uniquely identify all configuration items.***

ALC_CMC.5-5 The evaluator ***shall examine*** the configuration items to determine that they are identified in a way that is consistent with the CM documentation.

1007 Assurance that the CM system uniquely identifies all configuration items is gained by examining the identifiers for the configuration items. For both configuration items that comprise the TOE, and drafts of configuration items that are submitted by the developer as evaluation evidence, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

ALC_CMC.5.5C ***The CM system shall provide automated measures such that only authorised changes are made to the configuration items.***

ALC_CMC.5-6 The evaluator ***shall examine*** the CM access control measures described in the CM plan (cf. ALC_CMC.5.12C) to determine that they are automated and effective in preventing unauthorised access to the configuration items.

1008 The evaluator may use a number of methods to determine that the CM access control measures are effective. For example, the evaluator may exercise the access control measures to ensure that the procedures could not be bypassed. The evaluator may use the outputs generated by the CM system procedures required by ALC_CMC.5.16C. The evaluator may also witness a demonstration of the CM system to ensure that the access control measures employed are operating effectively.

ALC_CMC.5.6C ***The CM system shall support the production of the TOE by automated means.***

ALC_CMC.5-7 The evaluator ***shall check*** the CM plan (cf. ALC_CMC.5.12C) for automated procedures for supporting the production of the TOE.

1009 The term “production” applies to those processes adopted by the developer to progress the TOE from the implementation representation to a state acceptable for delivery to the end customer.

1010 The evaluator verifies the existence of automated production support procedures within the CM plan.

1011 The following are examples for automated means supporting the production of the TOE:

– a “make” tool (as provided with many software development tools) in the case of a software TOE;

– a tool ensuring automatically (for example by means of bar codes) that only parts are combined which indeed belong together in the case of a hardware TOE.

ALC_CMC.5-8 The evaluator ***shall examine*** the TOE production support procedures to determine that they are effective in ensuring that a TOE is generated that reflects its implementation representation.

1012 The production support procedures should describe which tools have to be used to produce the final TOE from the implementation representation in a clearly defined way. The conventions, directives, or other necessary constructs are described under ALC_TAT.

1013 The evaluator determines that by following the production support procedures the correct configuration items would be used to generate the TOE. For example, in a software TOE this may include checking that the automated production procedures ensure that all source files and related libraries are included in the compiled object code. Moreover, the procedures should ensure that compiler options and comparable other options are defined uniquely. For a hardware TOE, this work unit may include checking that the automatic production procedures ensure that the belonging parts are built together and no parts are missing.

1014 The customer can then be confident that the version of the TOE delivered for installation is derived from the implementation representation in an unambiguous way and implements the SFRs as described in the ST.

1015 The evaluator should bear in mind that the CM system need not necessarily possess the capability to produce the TOE, but should provide support for the process that will help reduce the probability of human error.

ALC_CMC.5.7C ***The CM system shall ensure that the person responsible for accepting a configuration item into CM is not the person who developed it.***

ALC_CMC.5-9 The evaluator ***shall examine*** the CM system to determine that it ensures that the person responsible for accepting a configuration item is not the person who developed it.

1016 The acceptance procedures describe who is responsible for accepting a configuration item. From these descriptions, the evaluator should be able to determine that the person who developed a configuration item is in no case responsible for its acceptance.

ALC_CMC.5.8C ***The CM system shall identify the configuration items that comprise the TSF.***

ALC_CMC.5-10 The evaluator ***shall examine*** the CM system to determine that it identifies the configuration items that comprise the TSF.

1017 The CM documentation should describe how the CM system identifies the configuration items that comprise the TSF. The evaluator should select a sample of configuration items covering each type of items, particularly containing TSF and non-TSF items, and check that they are correctly classified by the CM system.

1018 For guidance on sampling see A.2, Sampling.

ALC_CMC.5.9C ***The CM system shall support the audit of all changes to the TOE by automated means, including the originator, date, and time in the audit trail.***

ALC_CMC.5-11 The evaluator ***shall examine*** the CM system to determine that it supports the audit of all changes to the TOE by automated means, including the originator, date, and time in the audit trail.

1019 The evaluator should inspect a sample of audit trails and check, if they contain the minimum information.

ALC_CMC.5.10C ***The CM system shall provide an automated means to identify all other configuration items that are affected by the change of a given configuration item.***

ALC_CMC.5-12 The evaluator ***shall examine*** the CM system to determine that it provides an automated means to identify all other configuration items that are affected by the change of a given configuration item.

1020 The CM documentation should describe how the CM system identifies all other configuration items that are affected by the change of a given configuration item. The evaluator should select a sample of configuration items, covering all types of items, and exercise the automated means to determine that it identifies all items that are affected by the change of the selected item.

1021 For guidance on sampling see A.2, Sampling.

ALC_CMC.5.11C ***The CM system shall be able to identify the version of the implementation representation from which the TOE is generated.***

ALC_CMC.5-13 The evaluator ***shall examine*** the CM system to determine that it is able to identify the version of the implementation representation from which the TOE is generated.

1022 The CM documentation should describe how the CM system identifies the version of the implementation representation from which the TOE is generated. The evaluator should select a sample of the parts used to produce the TOE and should apply the CM system to verify that it identifies the corresponding implementation representation in the correct version.

1023 For guidance on sampling see A.2, Sampling.

ALC_CMC.5.12C ***The CM documentation shall include a CM plan.***

ALC_CMC.5-14 The evaluator ***shall check*** that the CM documentation provided includes a CM plan. The CM plan needs not to be a connected document, but it is recommended that there is a single document that describes where the various parts of the CM plan can be found. If the CM plan is no single document, the list in the following work unit gives hints regarding which context is expected.

ALC_CMC.5.13C ***The CM plan shall describe how the CM system is used for the development of the TOE.***

ALC_CMC.5-15 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used for the development of the TOE.

1024 The descriptions contained in a CM plan include, if applicable:

a) all activities performed in the TOE development that are subject to configuration management procedures (e.g. creation, modification or deletion of a configuration item, data-backup, archiving);

b) which means (e.g. CM tools, forms) have to be made available;

c) the usage of the CM tools: the necessary details for a user of the CM system to be able to operate the CM tools correctly in order to maintain the integrity of the TOE;

d) the production support procedures;

e) which other objects (development components, tools, assessment environments, etc) are taken under CM control;

f) the roles and responsibilities of individuals required to perform operations on individual configuration items (different roles may be identified for different types of configuration items (e.g. design documentation or source code));

g) how CM instances (e.g. change control boards, interface control working groups) are introduced and staffed;

h) the description of the change management;

i) the procedures that are used to ensure that only authorised individuals can make changes to configuration items;

j) the procedures that are used to ensure that concurrency problems do not occur as a result of simultaneous changes to configuration items;

k) the evidence that is generated as a result of application of the procedures. For example, for a change to a configuration item, the

CM system might record a description of the change, accountability for the change, identification of all configuration items affected, status (e.g. pending or completed), and date and time of the change. This might be recorded in an audit trail of changes made or change control records;

l) the approach to version control and unique referencing of TOE versions (e.g. covering the release of patches in operating systems, and the subsequent detection of their application).

ALC_CMC.5.14C ***The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.***

ALC_CMC.5-16 The evaluator ***shall examine*** the CM plan to determine that it describes the procedures used to accept modified or newly created configuration items as parts of the TOE.

1025 The descriptions of the acceptance procedures in the CM plan should include the developer roles or individuals responsible for the acceptance and the criteria to be used for acceptance. They should take into account all acceptance situations that may occur, in particular:

a) accepting an item into the CM system for the first time, in particular inclusion of software, firmware and hardware components from other manufacturers into the TOE (“integration”);

b) moving configuration items to the next life-cycle phase at each stage of the construction of the TOE (e.g. module, subsystem, system);

c) subsequent to transports between different development sites.

ALC_CMC.5.15C ***The evidence shall demonstrate that all configuration items are being maintained under the CM system.***

ALC_CMC.5-17 The evaluator ***shall check*** that the configuration items identified in the configuration list are being maintained by the CM system.

1026 The CM system employed by the developer should maintain the integrity of the TOE. The evaluator should check that for each type of configuration item (e.g. design documents or source code modules) contained in the configuration list there are examples of the evidence generated by the procedures described in the CM plan. In this case, the approach to sampling will depend upon the level of granularity used in the CM system to control CM items. Where, for example, 10,000 source code modules are identified in the configuration list, a different sampling strategy should be applied compared to the case in which there are only 5, or even 1. The emphasis of this activity should be on ensuring that the CM system is being operated correctly, rather than on the detection of any minor error.

1027 For guidance on sampling see A.2, Sampling.

ALC_CMC.5.16C ***The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.***

ALC_CMC.5-18 The evaluator ***shall check*** the CM documentation to ascertain that it includes the CM system records identified by the CM plan.

1028 The output produced by the CM system should provide the evidence that the evaluator needs to be confident that the CM plan is being applied, and also that all configuration items are being maintained by the CM system as required by ALC_CMC.5.15C. Example output could include change control forms, or configuration item access approval forms.

ALC_CMC.5-19 The evaluator ***shall examine*** the evidence to determine that the CM system is being operated in accordance with the CM plan.

1029 The evaluator should select and examine a sample of evidence covering each type of CM-relevant operation that has been performed on a configuration item (e.g. creation, modification, deletion, reversion to an earlier version) to confirm that all operations of the CM system have been carried out in line with documented procedures. The evaluator confirms that the evidence includes all the information identified for that operation in the CM plan. Examination of the evidence may require access to a CM tool that is used. The evaluator may choose to sample the evidence.

1030 For guidance on sampling see A.2, Sampling.

1031 Further confidence in the correct operation of the CM system and the effective maintenance of configuration items may be established by means of interview with selected development staff. In conducting such interviews, the evaluator should aim to gain a deeper understanding of how the CM system is used in practise as well as to confirm that the CM procedures are being applied as described in the CM documentation. Note that such interviews should complement rather than replace the examination of documentary evidence, and may not be necessary if the documentary evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and records alone. This is one case where clarification may be necessary through interviews.

1032 It is expected that the evaluator will visit the development site in support of this activity.

1033 For guidance on site visits see A.4, Site Visits.

13.2.5.4 Action ALC_CMC.5.2E

ALC_CMC.5-20 The evaluator ***shall examine*** the production support procedures to determine that by following these procedures a TOE would be produced like that one provided by the developer for testing activities.

1034 If the TOE is a small software TOE and production consists of compiling and linking, the evaluator might confirm the adequacy of the production support procedures by reapplying them himself.

1035 If the production process of the TOE is more complicated (as for example in the case of a smart card), but has already started, the evaluator should inspect the application of the production support procedures during a visit of the development site. He might compare a copy of the TOE produced in his presence with the samples used for his testing activities.

1036 For guidance on site visits see A.4, Site Visits.

1037 Otherwise the evaluator's determination should be based on the documentary evidence provided by the developer.

1038 This work unit may be performed in conjunction with the evaluation activities under Implementation representation (ADV_IMP).

13.3 CM scope (ALC_CMS)

13.3.1 Evaluation of sub-activity (ALC_CMS.1)

13.3.1.1 Objectives

1039 The objective of this sub-activity is to determine whether the developer performs configuration management on the TOE and the evaluation evidence.

13.3.1.2 Input

1040 The evaluation evidence for this sub-activity is:

a) the ST;

b) the configuration list.

13.3.1.3 Action ALC_CMS.1.1E

ALC_CMS.1.1C ***The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.***

ALC_CMS.1-1 The evaluator ***shall check*** that the configuration list includes the following set of items:

a) the TOE itself;

b) the evaluation evidence required by the SARs in the ST.

ALC_CMS.1.2C ***The configuration list shall uniquely identify the configuration items.***

ALC_CMS.1-2 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each configuration item.

1041 The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

13.3.2 Evaluation of sub-activity (ALC_CMS.2)

13.3.2.1 Objectives

1042 The objective of this sub-activity is to determine whether the configuration list includes the TOE, the parts that comprise the TOE, and the evaluation evidence.

13.3.2.2 Input

1043 The evaluation evidence for this sub-activity is:

a) the ST;

b) the configuration list.

13.3.2.3 Action ALC_CMS.2.1E

ALC_CMS.2.1C ***The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.***

ALC_CMS.2-1 The evaluator ***shall check*** that the configuration item list includes the set of items required by the CC.

1044 The list includes at least the following:

a) the TOE itself;

b) the parts that comprise the TOE;

c) the evaluation evidence required by the SARs.

ALC_CMS.2.2C ***The configuration list shall uniquely identify the configuration items.***

ALC_CMS.2-2 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each configuration item.

1045 The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

ALC_CMS.2.3C ***For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.***

ALC_CMS.2-3 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant configuration item.

1046 If only one developer is involved in the development of the TOE, this work unit is not applicable, and is therefore considered to be satisfied.

13.3.3 Evaluation of sub-activity (ALC_CMS.3)

13.3.3.1 Objectives

1047 The objective of this sub-activity is to determine whether the configuration list includes the TOE, the parts that comprise the TOE, the TOE implementation representation, and the evaluation evidence.

13.3.3.2 Input

1048 The evaluation evidence for this sub-activity is:

a) the ST;

b) the configuration list.

13.3.3.3 Action ALC_CMS.3.1E

ALC_CMS.3.1C ***The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.***

ALC_CMS.3-1 The evaluator ***shall check*** that the configuration list includes the following set of items:

a) the TOE itself;

b) the parts that comprise the TOE;

c) the TOE implementation representation;

d) the evaluation evidence required by the SARs in the ST.

ALC_CMS.3.2C ***The configuration list shall uniquely identify the configuration items.***

ALC_CMS.3-2 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each configuration item.

1049 The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

ALC_CMS.3.3C ***For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.***

ALC_CMS.3-3 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant configuration item.

1050 If only one developer is involved in the development of the TOE, this work unit is not applicable, and is therefore considered to be satisfied.

13.3.4 Evaluation of sub-activity (ALC_CMS.4)

13.3.4.1 Objectives

1051 The objective of this sub-activity is to determine whether the configuration list includes the TOE, the parts that comprise the TOE, the TOE implementation representation, security flaws, and the evaluation evidence.

13.3.4.2 Input

1052 The evaluation evidence for this sub-activity is:

a) the ST;

b) the configuration list.

13.3.4.3 Action ALC_CMS.4.1E

ALC_CMS.4.1C ***The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.***

ALC_CMS.4-1 The evaluator ***shall check*** that the configuration list includes the following set of items:

a) the TOE itself;

b) the parts that comprise the TOE;

c) the TOE implementation representation;

d) the evaluation evidence required by the SARs in the ST;

e) the documentation used to record details of reported security flaws associated with the implementation (e.g., problem status reports derived from a developer's problem database).

ALC_CMS.4.2C ***The configuration list shall uniquely identify the configuration items.***

ALC_CMS.4-2 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each configuration item.

1053 The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

ALC_CMS.4.3C ***For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.***

ALC_CMS.4-3 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant configuration item.

1054 If only one developer is involved in the development of the TOE, this work unit is not applicable, and is therefore considered to be satisfied.

13.3.5 Evaluation of sub-activity (ALC_CMS.5)

13.3.5.1 Objectives

1055 The objective of this sub-activity is to determine whether the configuration list includes the TOE, the parts that comprise the TOE, the TOE

implementation representation, security flaws, development tools and related information, and the evaluation evidence.

13.3.5.2 Input

1056 The evaluation evidence for this sub-activity is:

a) the ST;

b) the configuration list.

13.3.5.3 Action ALC_CMS.5.1E

ALC_CMS.5.1C ***The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; security flaw reports and resolution status; and development tools and related information.***

ALC_CMS.5-1 The evaluator ***shall check*** that the configuration list includes the following set of items:

a) the TOE itself;

b) the parts that comprise the TOE;

c) the TOE implementation representation;

d) the evaluation evidence required by the SARs in the ST;

e) the documentation used to record details of reported security flaws associated with the implementation (e.g., problem status reports derived from a developer's problem database);

f) all tools (incl. test software, if applicable) involved in the development and production of the TOE including the names, versions, configurations and roles of each development tool, and related documentation. E.g. for a software TOE: “development tools” are usually programming languages and compiler and “related documentation” comprises compiler and linker options. For a hardware TOE, “development tools” might be hardware design languages, simulation and synthesis tools, compilers, and “related documentation” might comprise compiler options again.

ALC_CMS.5.2C ***The configuration list shall uniquely identify the configuration items.***

ALC_CMS.5-2 The evaluator ***shall examine*** the configuration list to determine that it uniquely identifies each configuration item.

1057 The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration

items, and the correct version of each item, have been used during the evaluation.

ALC_CMS.5.3C ***For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.***

ALC_CMS.5-3 The evaluator ***shall check*** that the configuration list indicates the developer of each TSF relevant configuration item.

1058 If only one developer is involved in the development of the TOE, this work unit is not applicable, and is therefore considered to be satisfied.

13.4 Delivery (ALC_DEL)

13.4.1 Evaluation of sub-activity (ALC_DEL.1)

13.4.1.1 Objectives

1059 The objective of this sub-activity is to determine whether the delivery documentation describes all procedures used to maintain security of the TOE when distributing the TOE to the user.

13.4.1.2 Input

1060 The evaluation evidence for this sub-activity is:

a) the ST;

b) the delivery documentation.

13.4.1.3 Action ALC_DEL.1.1E

ALC_DEL.1.1C ***The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.***

ALC_DEL.1-1 The evaluator ***shall examine*** the delivery documentation to determine that it describes all procedures that are necessary to maintain security when distributing versions of the TOE or parts of it to the consumer.

1061 The delivery documentation describes proper procedures to maintain security of the TOE during transfer of the TOE or its component parts and to determine the identification of the TOE.

1062 The delivery documentation should cover the entire TOE, but may contain different procedures for different parts of the TOE. The evaluation should consider the totality of procedures.

1063 The delivery procedures should be applicable across all phases of delivery from the production environment to the installation environment (e.g. packaging, storage and distribution). Standard commercial practise for packaging and delivery may be acceptable. This includes shrink wrapped packaging, a security tape or a sealed envelope. For the distribution, physical (e.g. public mail or a private distribution service) or electronic (e.g. electronic mail or downloading off the Internet) procedures may be used.

1064 Cryptographic checksums or a software signature may be used by the developer to ensure that tampering or masquerading can be detected. Tamper proof seals additionally indicate if the confidentiality has been broken. For software TOEs, confidentiality might be assured by using encryption. If availability is of concern, a secure transportation might be required.

1065 Interpretation of the term “necessary to maintain security” will need to consider:

- The nature of the TOE (e.g. whether it is software or hardware).
- The overall security level stated for the TOE by the chosen level of the Vulnerability Assessment. If the TOE is required to be resistant against attackers of a certain potential in its intended environment, this should also apply to the delivery of the TOE. The evaluator should determine that a balanced approach has been taken, such that delivery does not present a weak point in an otherwise secure development process.
- The security objectives provided by the ST. In particular, the security aspects (integrity, confidentiality, availability) relevant for the actual TOE should be derived from the security objectives for the development environment defined in the ST. The emphasis in the delivery documentation is likely to be on measures related to integrity, as integrity of the TOE is always important. However, confidentiality and availability of the delivery will be of concern in the delivery of some TOEs; procedures relating to these aspects of the secure delivery should also be discussed in the procedures.

13.4.1.4 Implied evaluator action

ALC_DEL.1.2D ***The developer shall use the delivery procedures.***

ALC_DEL.1-2 The evaluator ***shall examine*** aspects of the delivery process to determine that the delivery procedures are used.

1066 The approach taken by the evaluator to check the application of delivery procedures will depend on the nature of the TOE, and the delivery process itself. In addition to examination of the procedures themselves, the evaluator seeks some assurance that they are applied in practise. Some possible approaches are:

a) a visit to the distribution site(s) where practical application of the procedures may be observed;

b) examination of the TOE at some stage during delivery, or after the user has received it (e.g. checking for tamper proof seals);

c) observing that the process is applied in practise when the evaluator obtains the TOE through regular channels;

d) questioning end users as to how the TOE was delivered.

1067 For guidance on site visits see A.4, Site Visits.

1068 It may be the case of a newly developed TOE that the delivery procedures have yet to be exercised. In these cases, the evaluator has to be satisfied that appropriate procedures and facilities are in place for future deliveries and that all personnel involved are aware of their responsibilities. The evaluator may request a “dry run” of a delivery if this is practical. If the developer has produced other similar products, then an examination of procedures in their use may be useful in providing assurance.

13.5 Development security (ALC_DVS)

13.5.1 Evaluation of sub-activity (ALC_DVS.1)

13.5.1.1 Objectives

1069 The objective of this sub-activity is to determine whether the developer's security controls on the development environment are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure that secure operation of the TOE is not compromised.

13.5.1.2 Input

1070 The evaluation evidence for this sub-activity is:

a) the ST;

b) the development security documentation.

1071 In addition, the evaluator may need to examine other deliverables to determine that the security controls are well-defined and followed. Specifically, the evaluator may need to examine the developer's configuration management documentation (the input for the Evaluation of sub-activity (ALC_CMC.4) “Production support and acceptance procedures” and the Evaluation of sub-activity (ALC_CMS.4) “Problem tracking CM coverage”). Evidence that the procedures are being applied is also required.

13.5.1.3 Action ALC_DVS.1.1E

ALC_DVS.1.1C ***The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.***

ALC_DVS.1-1 The evaluator ***shall examine*** the development security documentation to determine that it details all security measures used in the development environment that are necessary to protect the confidentiality and integrity of the TOE design and implementation.

1072 The evaluator determines what is necessary by first referring to the ST for any information that may assist in the determination of necessary protection, especially the security objectives for the development environment.

1073 If no explicit information is available from the ST the evaluator will need to make a determination of the necessary measures. In cases where the developer's measures are considered less than what is necessary, a clear justification should be provided for the assessment, based on a potential exploitable vulnerability.

1074 The following types of security measures are considered by the evaluator when examining the documentation:

a) physical, for example physical access controls used to prevent unauthorised access to the TOE development environment (during normal working hours and at other times);

b) procedural, for example covering:

- granting of access to the development environment or to specific parts of the environment such as development machines
- revocation of access rights when a person leaves the development team
- transfer of protected material within and out of the development environment and between different development sites in accordance with defined acceptance procedures
- admitting and escorting visitors to the development environment
- roles and responsibilities in ensuring the continued application of security measures, and the detection of security breaches.

c) personnel, for example any controls or checks made to establish the trustworthiness of new development staff;

d) other security measures, for example the logical protections on any development machines.

1075 The development security documentation should identify the locations at which development occurs, and describe the aspects of development performed, along with the security measures applied at each location and for transports between different locations. For example, development could occur at multiple facilities within a single building, multiple buildings at the same site, or at multiple sites. Transports of parts of the TOE or the unfinished TOE between different development sites are to be covered by Development security (ALC_DVS), whereas the transport of the finished TOE to the consumer is dealt with in Delivery (ALC_DEL).

1076 Development includes the production of the TOE.

ALC_DVS.1-2 The evaluator ***shall examine*** the development confidentiality and integrity policies in order to determine the sufficiency of the security measures employed.

1077 These include the policies governing:

a) what information relating to the TOE development needs to be kept confidential, and which members of the development staff are allowed to access such material;

b) what material must be protected from unauthorised modification in order to preserve the integrity of the TOE, and which members of the development staff are allowed to modify such material.

1078 The evaluator should determine that these policies are described in the development security documentation, that the security measures employed are consistent with the policies, and that they are complete.

1079 It should be noted that configuration management procedures will help protect the integrity of the TOE and the evaluator should avoid overlap with the work-units conducted for the CM capabilities (ALC_CMC). For example, the CM documentation may describe the security procedures necessary for controlling the roles or individuals who should have access to the development environment and who may modify the TOE.

1080 Whereas the CM capabilities (ALC_CMC) requirements are fixed, those for the Development security (ALC_DVS), mandating only necessary measures, are dependent on the nature of the TOE, and on information that may be provided in the ST. For example, the ST may identify a security objective for the development environment that requires the TOE to be developed by staff that has security clearance. The evaluators would then determine that such a policy had been applied under this sub-activity.

13.5.1.4 Action ALC_DVS.1.2E

ALC_DVS.1-3 The evaluator ***shall examine*** the development security documentation and associated evidence to determine that the security measures are being applied.

1081 This work unit requires the evaluator to determine that the security measures described in the development security documentation are being followed, such that the integrity of the TOE and the confidentiality of associated documentation is being adequately protected. For example, this could be determined by examination of the documentary evidence provided. Documentary evidence should be supplemented by visiting the development environment. A visit to the development environment will allow the evaluator to:

a) observe the application of security measures (e.g. physical measures);

b) examine documentary evidence of application of procedures;

c) interview development staff to check awareness of the development security policies and procedures, and their responsibilities.

1082 A development site visit is a useful means of gaining confidence in the measures being used. Any decision not to make such a visit should be determined in consultation with the evaluation authority.

1083 For guidance on site visits see A.4, Site Visits.

13.5.2 Evaluation of sub-activity (ALC_DVS.2)

13.5.2.1 Objectives

1084 The objective of this sub-activity is to determine whether the developer's security controls on the development environment are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure that secure operation of the TOE is not compromised. Additionally, sufficiency of the measures as applied is intended be justified.

13.5.2.2 Input

1085 The evaluation evidence for this sub-activity is:

a) the ST;

b) the development security documentation.

1086 In addition, the evaluator may need to examine other deliverables to determine that the security controls are well-defined and followed. Specifically, the evaluator may need to examine the developer's configuration management documentation (the input for the Evaluation of sub-activity (ALC_CMC.4) “Production support and acceptance procedures” and the Evaluation of sub-activity (ALC_CMS.4) “Problem tracking CM coverage”). Evidence that the procedures are being applied is also required.

13.5.2.3 Action ALC_DVS.2.1E

ALC_DVS.2.1C ***The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.***

ALC_DVS.2-1 The evaluator ***shall examine*** the development security documentation to determine that it details all security measures used in the development environment that are necessary to protect the confidentiality and integrity of the TOE design and implementation.

1087 The evaluator determines what is necessary by first referring to the ST for any information that may assist in the determination of necessary protection, especially the security objectives for the development environment.

1088 If no explicit information is available from the ST the evaluator will need to make a determination of the necessary measures. In cases where the developer's measures are considered less than what is necessary, a clear justification should be provided for the assessment, based on a potential exploitable vulnerability.

1089 The following types of security measures are considered by the evaluator when examining the documentation:

a) physical, for example physical access controls used to prevent unauthorised access to the TOE development environment (during normal working hours and at other times);

b) procedural, for example covering:

- granting of access to the development environment or to specific parts of the environment such as development machines
- revocation of access rights when a person leaves the development team
- transfer of protected material out of the development environment and between different development sites in accordance with defined acceptance procedures
- admitting and escorting visitors to the development environment
- roles and responsibilities in ensuring the continued application of security measures, and the detection of security breaches.

c) personnel, for example any controls or checks made to establish the trustworthiness of new development staff;

d) other security measures, for example the logical protections on any development machines.

1090 The development security documentation should identify the locations at which development occurs, and describe the aspects of development performed, along with the security measures applied at each location and for transports between different locations. For example, development could occur at multiple facilities within a single building, multiple buildings at the same site, or at multiple sites. Transports of parts of the TOE or the unfinished TOE between different development sites are to be covered by the Development security (ALC_DVS), whereas the transport of the finished TOE to the consumer is dealt with in the Delivery (ALC_DEL).

1091 Development includes the production of the TOE.

ALC_DVS.2.2C ***The development security documentation shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.***

ALC_DVS.2-2 The evaluator ***shall examine*** the development security documentation to determine that an appropriate justification is given why the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

1092 Since attacks on the TOE or its related information are assumed in different design and production stages, measures and procedures need to have an appropriate level necessary to prevent those attacks or to make them more difficult.

1093 Since this level depends on the overall attack potential claimed for the TOE (cf. the Vulnerability analysis (AVA_VAN) component chosen), the development security documentation should justify the necessary level of protection to maintain the confidentiality and integrity of the TOE. This level has to be achieved by the security measures applied.

1094 The concept of protection measures should be consistent, and the justification should include an analysis of how the measures are mutually supportive. All aspects of development and production on all the different sites with all roles involved up to delivery of the TOE should be analysed.

1095 Justification may include an analysis of potential vulnerabilities taking the applied security measures into account.

1096 There may be a convincing argument showing that e.g.

– The technical measures and mechanisms of the developer's infrastructure are sufficient for keeping the appropriate security level (e.g. cryptographic mechanisms as well as physical protection mechanisms, properties of the CM system (cf. ALC_CMC.4-5));

– The system containing the implementation representation of the TOE (including concerning guidance documents) provides effective protection against logical attacks e.g. by “Trojan” code or viruses. It might be adequate, if the implementation representation is kept on an isolated system where only the software necessary to maintain it is installed and where no additional software is installed afterwards.

– Data brought into this system should be carefully considered to prevent the installation of hidden functionality onto the system. The effectiveness of these measures should be tested, e.g. by independently trying to get access to the machine, install some additional executable (program, macro etc.) or get some information out of the machine using logical attacks.

– The appropriate organisational (procedural and personal) measures are unconditionally enforced.

ALC_DVS.2-3 The evaluator ***shall examine*** the development confidentiality and integrity policies in order to determine the sufficiency of the security measures employed.

1097 These include the policies governing:

a) what information relating to the TOE development needs to be kept confidential, and which members of the development staff are allowed to access such material;

b) what material must be protected from unauthorised modification in order to preserve the integrity of the TOE, and which members of the development staff are allowed to modify such material.

1098 The evaluator should determine that these policies are described in the development security documentation, that the security measures employed are consistent with the policies, and that they are complete.

1099 It should be noted that configuration management procedures will help protect the integrity of the TOE and the evaluator should avoid overlap with the work-units conducted for the CM capabilities (ALC_CMC). For example, the CM documentation may describe the security procedures necessary for controlling the roles or individuals who should have access to the development environment and who may modify the TOE.

1100 Whereas the CM capabilities (ALC_CMC) requirements are fixed, those for the Development security (ALC_DVS), mandating only necessary measures, are dependent on the nature of the TOE, and on information that may be provided in the ST. For example, the ST may identify a security objective for the development environment that requires the TOE to be developed by staff that has security clearance. The evaluators would then determine that such a policy had been applied under this sub-activity.

13.5.2.4 Action ALC_DVS.2.2E

ALC_DVS.2-4 The evaluator ***shall examine*** the development security documentation and associated evidence to determine that the security measures are being applied.

1101 This work unit requires the evaluator to determine that the security measures described in the development security documentation are being followed, such that the integrity of the TOE and the confidentiality of associated documentation is being adequately protected. For example, this could be determined by examination of the documentary evidence provided. Documentary evidence should be supplemented by visiting the development environment. A visit to the development environment will allow the evaluator to:

a) observe the application of security measures (e.g. physical measures);

b) examine documentary evidence of application of procedures;

c) interview development staff to check awareness of the development security policies and procedures, and their responsibilities.

1102 A development site visit is a useful means of gaining confidence in the measures being used. Any decision not to make such a visit should be determined in consultation with the evaluation authority.

1103 For guidance on site visits see A.4, Site Visits.

13.6 Flaw remediation (ALC_FLR)

13.6.1 Evaluation of sub-activity (ALC_FLR.1)

13.6.1.1 Objectives

1104 The objective of this sub-activity is to determine whether the developer has established flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to TOE users.

13.6.1.2 Input

1105 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation.

13.6.1.3 Action ALC_FLR.1.1E

ALC_FLR.1.1C ***The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.***

ALC_FLR.1-1 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes the procedures used to track all reported security flaws in each release of the TOE.

1106 The procedures describe the actions that are taken by the developer from the time each suspected security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame, from initial detection through ascertaining that the flaw is a security flaw, to resolution of the security flaw.

1107 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only that there be an explanation of why the flaw is not security-relevant.

1108 While these requirements do not mandate that there be a publicised means for TOE users to report security flaws, they do mandate that all security flaws that are reported be tracked. That is, a reported security flaw cannot be ignored simply because it comes from outside the developer's organisation.

ALC_FLR.1.2C ***The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.***

ALC_FLR.1-2 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would produce a description of each security flaw in terms of its nature and effects.

1109 The procedures identify the actions that are taken by the developer to describe the nature and effects of each security flaw in sufficient detail to be

able to reproduce it. The description of the nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's effects identifies the portions of the TSF that are affected and how those portions are affected. For example, a security flaw in the implementation might be found that affects the identification and authentication enforced by the TSF by permitting authentication with the password “BACK DOOR”.

ALC_FLR.1-3 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would identify the status of finding a correction to each security flaw.

1110 The flaw remediation procedures identify the different stages of security flaws. This differentiation includes at least: suspected security flaws that have been reported, suspected security flaws that have been confirmed to be security flaws, and security flaws whose solutions have been implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet investigated, flaws that are under investigation, security flaws for which a solution has been found but not yet implemented) be included.

ALC_FLR.1.3C ***The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.***

ALC_FLR.1-4 The evaluator ***shall check*** the flaw remediation procedures to determine that the application of these procedures would identify the corrective action for each security flaw.

1111 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes both those measures serving as only an interim solution (until the repair is issued) as well as those serving as a permanent solution (where it is determined that the procedural measure is the best solution).

1112 If the source of the security flaw is a documentation error, the corrective action consists of an update of the affected TOE guidance. If the corrective action is a procedural measure, this measure will include an update made to the affected TOE guidance to reflect these corrective procedures.

ALC_FLR.1.4C ***The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.***

ALC_FLR.1-5 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes a means of providing the TOE users with the necessary information on each security flaw.

1113 The *necessary information* about each security flaw consists of its description (not necessarily at the same level of detail as that provided as part

of work unit ALC_FLR.1-2), the prescribed corrective action, and any associated guidance on implementing the correction.

1114 TOE users may be provided with such information, correction, and documentation updates in any of several ways, such as their posting to a website, their being sent to TOE users, or arrangements made for the developer to install the correction. In cases where the means of providing this information requires action to be initiated by the TOE user, the evaluator examines any TOE guidance to ensure that it contains instructions for retrieving the information.

1115 The only metric for assessing the adequacy of the method used for providing the information, corrections and guidance is that there be a reasonable expectation that TOE users can obtain or receive it. For example, consider the method of dissemination where the requisite data is posted to a website for one month, and the TOE users know that this will happen and when this will happen. This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet it is feasible that the TOE user could obtain the necessary information. On the other hand, if the information were posted to the website for only one hour, yet TOE users had no way of knowing this or when it would be posted, it is infeasible that they would ever get the necessary information.

13.6.2 Evaluation of sub-activity (ALC_FLR.2)

13.6.2.1 Objectives

1116 The objective of this sub-activity is to determine whether the developer has established flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to TOE users. Additionally, this sub-activity determines whether the developer's procedures provide for the corrections of security flaws, for the receipt of flaw reports from TOE users, and for assurance that the corrections introduce no new security flaws.

1117 In order for the developer to be able to act appropriately upon security flaw reports from TOE users, TOE users need to understand how to submit security flaw reports to the developer, and developers need to know how to receive these reports. Flaw remediation guidance addressed to the TOE user ensures that TOE users are aware of how to communicate with the developer; flaw remediation procedures describe the developer's role is such communication

13.6.2.2 Input

1118 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation;

b) flaw remediation guidance documentation.

13.6.2.3 Action ALC_FLR.2.1E

ALC_FLR.2.1C ***The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.***

ALC_FLR.2-1 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes the procedures used to track all reported security flaws in each release of the TOE.

1119 The procedures describe the actions that are taken by the developer from the time each suspected security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame, from initial detection through ascertaining that the flaw is a security flaw, to resolution of the security flaw.

1120 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only that there be an explanation of why the flaw is not security-relevant.

ALC_FLR.2.2C ***The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.***

ALC_FLR.2-2 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would produce a description of each security flaw in terms of its nature and effects.

1121 The procedures identify the actions that are taken by the developer to describe the nature and effects of each security flaw in sufficient detail to be able to reproduce it. The description of the nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's effects identifies the portions of the TSF that are affected and how those portions are affected. For example, a security flaw in the implementation might be found that affects the identification and authentication enforced by the TSF by permitting authentication with the password “BACKDOOR”.

ALC_FLR.2-3 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would identify the status of finding a correction to each security flaw.

1122 The flaw remediation procedures identify the different stages of security flaws. This differentiation includes at least: suspected security flaws that have been reported, suspected security flaws that have been confirmed to be security flaws, and security flaws whose solutions have been implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet investigated, flaws that are under investigation, security flaws for which a solution has been found but not yet implemented) be included.

ALC_FLR.2.3C ***The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.***

ALC_FLR.2-4 The evaluator ***shall check*** the flaw remediation procedures to determine that the application of these procedures would identify the corrective action for each security flaw.

1123 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes both those measures serving as only an interim solution (until the repair is issued) as well as those serving as a permanent solution (where it is determined that the procedural measure is the best solution).

1124 If the source of the security flaw is a documentation error, the corrective action consists of an update of the affected TOE guidance. If the corrective action is a procedural measure, this measure will include an update made to the affected TOE guidance to reflect these corrective procedures.

ALC_FLR.2.4C ***The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.***

ALC_FLR.2-5 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes a means of providing the TOE users with the necessary information on each security flaw.

1125 *The necessary information* about each security flaw consists of its description (not necessarily at the same level of detail as that provided as part of work unit ALC_FLR.2-2), the prescribed corrective action, and any associated guidance on implementing the correction.

1126 TOE users may be provided with such information, correction, and documentation updates in any of several ways, such as their posting to a website, their being sent to TOE users, or arrangements made for the developer to install the correction. In cases where the means of providing this information requires action to be initiated by the TOE user, the evaluator examines any TOE guidance to ensure that it contains instructions for retrieving the information.

1127 The only metric for assessing the adequacy of the method used for providing the information, corrections and guidance is that there be a reasonable expectation that TOE users can obtain or receive it. For example, consider the method of dissemination where the requisite data is posted to a website for one month, and the TOE users know that this will happen and when this will happen. This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet it is feasible that the TOE user could obtain the necessary information. On the other hand, if the information were posted to the website for only one hour, yet TOE users had no way of

knowing this or when it would be posted, it is infeasible that they would ever get the necessary information.

ALC_FLR.2.5C ***The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.***

ALC_FLR.2-6 The evaluator ***shall examine*** the flaw remediation procedures to determine that they describe procedures for the developer to accept reports of security flaws or requests for corrections to such flaws.

1128 The procedures ensure that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws. This means of contact may be part of a more general contact facility for reporting non-security related problems.

1129 The use of these procedures is not restricted to TOE users; however, only the TOE users are actively supplied with the details of these procedures. Others who might have access to or familiarity with the TOE can use the same procedures to submit reports to the developer, who is then expected to process them. Any means of submitting reports to the developer, other than those identified by the developer, are beyond the scope of this work unit; reports generated by other means need not be addressed.

ALC_FLR.2.6C ***The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.***

ALC_FLR.2-7 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure every reported flaw is corrected.

1130 The flaw remediation procedures cover not only those security flaws discovered and reported by developer personnel, but also those reported by TOE users. The procedures are sufficiently detailed so that they describe how it is ensured that each reported security flaw is corrected. The procedures contain reasonable steps that show progress leading to the eventual, inevitable resolution.

1131 The procedures describe the process that is taken from the point at which the suspected security flaw is determined to be a security flaw to the point at which it is resolved.

ALC_FLR.2-8 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure that the TOE users are issued remediation procedures for each security flaw.

1132 The procedures describe the process that is taken from the point at which a security flaw is resolved to the point at which the remediation procedures are

provided. The procedures for delivering corrective actions should be consistent with the security objectives; they need not necessarily be identical to the procedures used for delivering the TOE, as documented to meet ALC_DEL, if included in the assurance requirements. For example, if the hardware portion of a TOE were originally delivered by bonded courier, updates to hardware resulting from flaw remediation would likewise be expected to be distributed by bonded courier. Updates unrelated to flaw remediation would follow the procedures set forth in the documentation meeting the Delivery (ALC_DEL) requirements.

ALC_FLR.2.7C ***The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.***

ALC_FLR.2-9 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would result in safeguards that the potential correction contains no adverse effects.

1133 Through analysis, testing, or a combination of the two, the developer may reduce the likelihood that adverse effects will be introduced when a security flaw is corrected. The evaluator assesses whether the procedures provide detail in how the necessary mix of analysis and testing actions is to be determined for a given correction.

1134 The evaluator also determines that, for instances where the source of the security flaw is a documentation problem, the procedures include the means of safeguarding against the introduction of contradictions with other documentation.

ALC_FLR.2.8C ***The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.***

ALC_FLR.2-10 The evaluator ***shall examine*** the flaw remediation guidance to determine that the application of these procedures would result in a means for the TOE user to provide reports of suspected security flaws or requests for corrections to such flaws.

1135 The guidance ensures that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws.

13.6.3 Evaluation of sub-activity (ALC_FLR.3)

13.6.3.1 Objectives

1136 The objective of this sub-activity is to determine whether the developer has established flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to TOE users. Additionally, this sub-activity determines whether the developer's procedures provide for the corrections of

security flaws, for the receipt of flaw reports from TOE users, for assurance that the corrections introduce no new security flaws, for the establishment of a point of contact for each TOE user, and for the timely issue of corrective actions to TOE users.

1137 In order for the developer to be able to act appropriately upon security flaw reports from TOE users, TOE users need to understand how to submit security flaw reports to the developer, and developers need to know how to receive these reports. Flaw remediation guidance addressed to the TOE user ensures that TOE users are aware of how to communicate with the developer; flaw remediation procedures describe the developer's role is such communication.

13.6.3.2 Input

1138 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation;

b) flaw remediation guidance documentation.

13.6.3.3 Action ALC_FLR.3.1E

ALC_FLR.3.1C ***The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.***

ALC_FLR.3-1 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes the procedures used to track all reported security flaws in each release of the TOE.

1139 The procedures describe the actions that are taken by the developer from the time each suspected security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame, from initial detection through ascertaining that the flaw is a security flaw, to resolution of the security flaw.

1140 If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only that there be an explanation of why the flaw is not security-relevant.

ALC_FLR.3.2C ***The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.***

ALC_FLR.3-2 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would produce a description of each security flaw in terms of its nature and effects.

1141 The procedures identify the actions that are taken by the developer to describe the nature and effects of each security flaw in sufficient detail to be able to reproduce it. The description of the nature of a security flaw

addresses whether it is an error in the documentation, a flaw in the design of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's effects identifies the portions of the TSF that are affected and how those portions are affected. For example, a security flaw in the implementation might be found that affects the identification and authentication enforced by the TSF by permitting authentication with the password “BACKDOOR”.

ALC_FLR.3-3 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would identify the status of finding a correction to each security flaw.

1142 The flaw remediation procedures identify the different stages of security flaws. This differentiation includes at least: suspected security flaws that have been reported, suspected security flaws that have been confirmed to be security flaws, and security flaws whose solutions have been implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet investigated, flaws that are under investigation, security flaws for which a solution has been found but not yet implemented) be included.

ALC_FLR.3.3C ***The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.***

ALC_FLR.3-4 The evaluator ***shall check*** the flaw remediation procedures to determine that the application of these procedures would identify the corrective action for each security flaw.

1143 *Corrective action* may consist of a repair to the hardware, firmware, or software portions of the TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes both those measures serving as only an interim solution (until the repair is issued) as well as those serving as a permanent solution (where it is determined that the procedural measure is the best solution).

1144 If the source of the security flaw is a documentation error, the corrective action consists of an update of the affected TOE guidance. If the corrective action is a procedural measure, this measure will include an update made to the affected TOE guidance to reflect these corrective procedures.

ALC_FLR.3.4C ***The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.***

ALC_FLR.3-5 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes a means of providing the TOE users with the necessary information on each security flaw.

1145 *The necessary information* about each security flaw consists of its description (not necessarily at the same level of detail as that provided as part

of work unit ALC_FLR.3-2), the prescribed corrective action, and any associated guidance on implementing the correction.

1146 TOE users may be provided with such information, correction, and documentation updates in any of several ways, such as their posting to a website, their being sent to TOE users, or arrangements made for the developer to install the correction. In cases where the means of providing this information requires action to be initiated by the TOE user, the evaluator examines any TOE guidance to ensure that it contains instructions for retrieving the information.

1147 The only metric for assessing the adequacy of the method used for providing the information, corrections and guidance is that there be a reasonable expectation that TOE users can obtain or receive it. For example, consider the method of dissemination where the requisite data is posted to a website for one month, and the TOE users know that this will happen and when this will happen. This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet it is feasible that the TOE user could obtain the necessary information. On the other hand, if the information were posted to the website for only one hour, yet TOE users had no way of knowing this or when it would be posted, it is infeasible that they would ever get the necessary information.

1148 For TOE users who register with the developer (see work unit ALC_FLR.3-12), the passive availability of this information is not sufficient. Developers must actively send the information (or a notification of its availability) to registered TOE users.

ALC_FLR.3.5C ***The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.***

ALC_FLR.3-6 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would result in a means for the developer to receive from TOE user reports of suspected security flaws or requests for corrections to such flaws.

1149 The procedures ensure that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws. This means of contact may be part of a more general contact facility for reporting non-security related problems.

1150 The use of these procedures is not restricted to TOE users; however, only the TOE users are actively supplied with the details of these procedures. Others who might have access to or familiarity with the TOE can use the same procedures to submit reports to the developer, who is then expected to process them. Any means of submitting reports to the developer, other than those identified by the developer, are beyond the scope of this work unit; reports generated by other means need not be addressed.

ALC_FLR.3.6C ***The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.***

ALC_FLR.3-7 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would result in a timely means of providing the registered TOE users who might be affected with reports about, and associated corrections to, each security flaw.

1151 The issue of timeliness applies to the issuance of both security flaw reports and the associated corrections. However, these need not be issued at the same time. It is recognised that flaw reports should be generated and issued as soon as an interim solution is found, even if that solution is as drastic as turn off the TOE. Likewise, when a more permanent (and less drastic) solution is found, it should be issued without undue delay.

1152 It is unnecessary to restrict the recipients of the reports and associated corrections to only those TOE users who might be affected by the security flaw; it is permissible that all TOE users be given such reports and corrections for all security flaws, provided such is done in a timely manner.

ALC_FLR.3-8 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would result in automatic distribution of the reports and associated corrections to the registered TOE users who might be affected.

1153 *Automatic distribution* does not mean that human interaction with the distribution method is not permitted. In fact, the distribution method could consist entirely of manual procedures, perhaps through a closely monitored procedure with prescribed escalation upon the lack of issue of reports or corrections.

1154 It is unnecessary to restrict the recipients of the reports and associated corrections to only those TOE users who might be affected by the security flaw; it is permissible that all TOE users be given such reports and corrections for all security flaws, provided such is done automatically.

ALC_FLR.3.7C ***The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.***

ALC_FLR.3-9 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure that every reported flaw is remediated.

1155 The flaw remediation procedures cover not only those security flaws discovered and reported by developer personnel, but also those reported by TOE users. The procedures are sufficiently detailed so that they describe how it is ensured that each reported security flaw is remediated. The

procedures contain reasonable steps that show progress leading to the eventual, inevitable resolution.

1156 The procedures describe the process that is taken from the point at which the suspected security flaw is determined to be a security flaw to the point at which it is resolved.

ALC_FLR.3-10 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure that the TOE users are issued remediation procedures for each security flaw.

1157 The procedures describe the process that is taken from the point at which a security flaw is resolved to the point at which the remediation procedures are provided. The procedures for delivering remediation procedures should be consistent with the security objectives; they need not necessarily be identical to the procedures used for delivering the TOE, as documented to meet Delivery (ALC_DEL), if included in the assurance requirements. For example, if the hardware portion of a TOE were originally delivered by bonded courier, updates to hardware resulting from flaw remediation would likewise be expected to be distributed by bonded courier. Updates unrelated to flaw remediation would follow the procedures set forth in the documentation meeting the Delivery (ALC_DEL) requirements.

ALC_FLR.3.8C ***The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.***

ALC_FLR.3-11 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would result in safeguards that the potential correction contains no adverse effects.

1158 Through analysis, testing, or a combination of the two, the developer may reduce the likelihood that adverse effects will be introduced when a security flaw is corrected. The evaluator assesses whether the procedures provide detail in how the necessary mix of analysis and testing actions is to be determined for a given correction.

1159 The evaluator also determines that, for instances where the source of the security flaw is a documentation problem, the procedures include the means of safeguarding against the introduction of contradictions with other documentation.

ALC_FLR.3.9C ***The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.***

ALC_FLR.3-12 The evaluator ***shall examine*** the flaw remediation guidance to determine that the application of these procedures would result in a means for the TOE user to provide reports of suspected security flaws or requests for corrections to such flaws.

1160 The guidance ensures that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws.

ALC_FLR.3.10C ***The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.***

ALC_FLR.3-13 The evaluator ***shall examine*** the flaw remediation guidance to determine that it describes a means of enabling the TOE users to register with the developer.

1161 *Enabling the TOE users to register with the developer* simply means having a way for each TOE user to provide the developer with a point of contact; this point of contact is to be used to provide the TOE user with information related to security flaws that might affect that TOE user, along with any corrections to the security flaw. Registering the TOE user may be accomplished as part of the standard procedures that TOE users undergo to identify themselves to the developer, for the purposes of registering a software licence, or for obtaining update and other useful information.

1162 There need not be one registered TOE user per installation of the TOE; it would be sufficient if there were one registered TOE user for an organisation. For example, a corporate TOE user might have a centralised acquisition office for all of its sites. In this case, the acquisition office would be a sufficient point of contact for all of that TOE user's sites, so that all of the TOE user's installations of the TOE have a registered point of contact.

1163 In either case, it must be possible to associate each TOE that is delivered with an organisation in order to ensure that there is a registered user for each TOE. For organisations that have many different addresses, this assures that there will be no user who is erroneously presumed to be covered by a registered TOE user.

1164 It should be noted that TOE users need not register; they must only be provided with a means of doing so. However, users who choose to register must be directly sent the information (or a notification of its availability).

ALC_FLR.3.11C ***The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.***

ALC_FLR.3-14 The evaluator ***shall examine*** the flaw remediation guidance to determine that it identifies specific points of contact for user reports and enquiries about security issues involving the TOE.

1165 The guidance includes a means whereby registered TOE users can interact with the developer to report discovered security flaws in the TOE or to make enquiries regarding discovered security flaws in the TOE.

13.7 Life-cycle definition (ALC_LCD)

13.7.1 Evaluation of sub-activity (ALC_LCD.1)

13.7.1.1 Objectives

1166 The objective of this sub-activity is to determine whether the developer has used a documented model of the TOE life-cycle.

13.7.1.2 Input

1167 The evaluation evidence for this sub-activity is:

a) the ST;

b) the life-cycle definition documentation.

13.7.1.3 Action ALC_LCD.1.1E

ALC_LCD.1.1C ***The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.***

ALC_LCD.1-1 The evaluator ***shall examine*** the documented description of the life-cycle model used to determine that it covers the development and maintenance process.

1168 The description of the life-cycle model should include:

a) information on the life-cycle phases of the TOE and the boundaries between the subsequent phases;

b) information on the procedures, tools and techniques used by the developer (e.g. for design, coding, testing, bug-fixing);

c) overall management structure governing the application of the procedures (e.g. an identification and description of the individual responsibilities for each of the procedures required by the development and maintenance process covered by the life-cycle model);

d) information on which parts of the TOE are delivered by subcontractors, if subcontractors are involved.

1169 Evaluation of sub-activity (ALC_LCD.1) does not require the model used to conform to any standard life-cycle model.

ALC_LCD.1.2C ***The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.***

ALC_LCD.1-2 The evaluator ***shall examine*** the life-cycle model to determine that use of the procedures, tools and techniques described by the life-cycle model will make

the necessary positive contribution to the development and maintenance of the TOE.

1170 The information provided in the life-cycle model gives the evaluator assurance that the development and maintenance procedures adopted would minimise the likelihood of security flaws. For example, if the life-cycle model described the review process, but did not make provision for recording changes to components, then the evaluator may be less confident that errors will not be introduced into the TOE. The evaluator may gain further assurance by comparing the description of the model against an understanding of the development process gleaned from performing other evaluator actions relating to the TOE development (e.g. those covered under the CM capabilities (ALC_CMC)). Identified deficiencies in the life-cycle model will be of concern if they might reasonably be expected to give rise to the introduction of flaws into the TOE, either accidentally or deliberately.

1171 The CC does not mandate any particular development approach, and each should be judged on merit. For example, spiral, rapid-prototyping and waterfall approaches to design can all be used to produce a quality TOE if applied in a controlled environment.

13.7.2 Evaluation of sub-activity (ALC_LCD.2)

13.7.2.1 Objectives

1172 The objective of this sub-activity is to determine whether the developer has used a documented and measurable model of the TOE life-cycle.

13.7.2.2 Input

1173 The evaluation evidence for this sub-activity is:

a) the ST;

b) the life-cycle definition documentation;

c) information about the standard used;

d) the life-cycle output documentation.

13.7.2.3 Action ALC_LCD.2.1E

ALC_LCD.2.1C ***The life-cycle definition documentation shall describe the model used to develop and maintain the TOE, including the details of its arithmetic parameters and/or metrics used to measure the quality of the TOE and/or its development.***

ALC_LCD.2-1 The evaluator ***shall examine*** the documented description of the life-cycle model used to determine that it covers the development and maintenance process, including the details of its arithmetic parameters and/or metrics used to measure the TOE development.

1174 The description of the life-cycle model should include:

a) information on the life-cycle phases of the TOE and the boundaries between the subsequent phases;

b) information on the procedures, tools and techniques used by the developer (e.g. for design, coding, testing, bug-fixing);

c) overall management structure governing the application of the procedures (e.g. an identification and description of the individual responsibilities for each of the procedures required by the development and maintenance process covered by the life-cycle model);

d) information on which parts of the TOE are delivered by subcontractors, if subcontractors are involved;

e) information on the parameters/metrics that are used to measure the TOE development. Metrics standards typically include guides for measuring and producing reliable products and cover the aspects reliability, quality, performance, complexity and cost. For the evaluation all those metrics are of relevance, which are used to increase quality by decreasing the probability of faults and thereby in turn increase assurance in the security of the TOE.

ALC_LCD.2.2C ***The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.***

ALC_LCD.2-2 The evaluator ***shall examine*** the life-cycle model to determine that use of the procedures, tools and techniques described by the life-cycle model will make the necessary positive contribution to the development and maintenance of the TOE.

1175 The information provided in the life-cycle model gives the evaluator assurance that the development and maintenance procedures adopted would minimise the likelihood of security flaws. For example, if the life-cycle model described the review process, but did not make provision for recording changes to components, then the evaluator may be less confident that errors will not be introduced into the TOE. The evaluator may gain further assurance by comparing the description of the model against an understanding of the development process gleaned from performing other evaluator actions relating to the TOE development (e.g. those covered under the CM capabilities (ALC_CMC)). Identified deficiencies in the life-cycle model will be of concern if they might reasonably be expected to give rise to the introduction of flaws into the TOE, either accidentally or deliberately.

1176 The CC does not mandate any particular development approach, and each should be judged on merit. For example, spiral, rapid-prototyping and waterfall approaches to design can all be used to produce a quality TOE if applied in a controlled environment.

1177 For the metrics/measurements used in the life-cycle model, evidence has to be provided that shows how those metrics/measurements usefully contribute to the minimisation of the likelihood of flaws. This can be viewed as the overall goal for measurement in an ALC context. As a consequence the metrics/measurements have to be selected based on their capability to achieve that overall goal or contribute to that. In the first place a metric/measure is suitable with respect to ALC if a correlation between the metric/measure and the number of flaws can be stated with a certain degree of reliability. But also a metric/measure useful for management purposes as for planning and monitoring the TOE development are helpful since badly managed projects are endangered to produce bad quality and to introduce flaws.

1178 It may be possible to use metrics for quality improvement, for which this use is not obvious. For example a metric to estimate the expected cost of a product development may help quality, if the developer can show that this is used to provide an adequate budget for development projects and that this helps to avoid quality problems arising from resource shortages.

1179 It is not required that every single step in the life cycle of the TOE is measurable. However the evaluator should see from the description of the measures and procedures that the metrics are appropriate to control the overall quality of the TOE and to minimise possible security flaws by this.

ALC_LCD.2.3C ***The life-cycle output documentation shall provide the results of the measurements of the TOE development using the measurable life-cycle model.***

ALC_LCD.2-3 The evaluator ***shall examine*** the life-cycle output documentation to determine that it provides the results of the measurements of the TOE development using the measurable life-cycle model.

1180 The results of the measurements and the life-cycle progress of the TOE should be in accordance with the life-cycle model.

1181 The output documentation should not only include numeric values of the metrics but should also document actions taken as a result of the measurements and in accordance with the model. For example there may be a requirement that a certain design phase needs to be repeated, if some error rates measured during testing are outside of a defined threshold. In this case the documentation should show that such action was taken, if indeed the thresholds were not met.

1182 If the evaluation is conducted in parallel with the development of the TOE it may be possible that quality measurements have not been used in the past. In this case the evaluator should use the documentation of the planned procedures in order to gain confidence that corrective actions are defined if results of quality measurements deviate from some threshold.

13.8 Tools and techniques (ALC_TAT)

13.8.1 Evaluation of sub-activity (ALC_TAT.1)

13.8.1.1 Objectives

1183 The objective of this sub-activity is to determine whether the developer has used well-defined development tools (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results.

13.8.1.2 Input

1184 The evaluation evidence for this sub-activity is:

a) the development tool documentation;

b) the subset of the implementation representation.

13.8.1.3 Application notes

1185 This work may be performed in parallel with the evaluation activities under Implementation representation (ADV_IMP), specifically with regard to determining the use of features in the tools that will affect the object code (e.g. compilation options).

13.8.1.4 Action ALC_TAT.1.1E

ALC_TAT.1.1C ***Each development tool used for implementation shall be well-defined.***

ALC_TAT.1-1 The evaluator ***shall examine*** the development tool documentation provided to determine that each development tools is well-defined.

1186 For example, a well-defined language, compiler or CAD system may be considered to be one that conforms to a recognised standard, such as the ISO standards. A well-defined language is one that has a clear and complete description of its syntax, and a detailed description of the semantics of each construct.

ALC_TAT.1.2C ***The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.***

ALC_TAT.1-2 The evaluator ***shall examine*** the documentation of each development tool to determine that it unambiguously defines the meaning of all statements as well as all conventions and directives used in the implementation.

1187 The development tool documentation (e.g. programming language specifications and user manuals) should cover all statements used in the implementation representation of the TOE, and for each such statement should provide a clear and unambiguous definition of the purpose and effect of that statement. This work may be performed in parallel with the

evaluator's examination of the implementation representation performed during the ADV_IMP sub-activity. The key test the evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to be able to understand the implementation representation. The documentation should not assume (for example) that the reader is an expert in the programming language used.

1188 Reference to the use of a documented standard is an acceptable approach to meet this requirement, provided that the standard is available to the evaluator. Any differences from the standard should be documented.

1189 The critical test is whether the evaluator can understand the TOE source code when performing source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can additionally be used in searching for problem areas:

a) In the language definition, phrases such as “the effect of this construct is undefined” and terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.

b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a common source of ambiguity problems.

c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is often poorly defined.

1190 Most languages in common use, however well designed, will have some problematic constructs. If the implementation language is mostly well defined, but some problematic constructs exist, then an inconclusive verdict should be assigned, pending examination of the source code.

1191 The evaluator should verify, during the examination of source code, that any use of the problematic constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs precluded by the documented standard are not used.

1192 The development tool documentation should define all conventions and directives used in the implementation.

ALC_TAT.1.3C ***The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.***

ALC_TAT.1-3 The evaluator ***shall examine*** the development tool documentation to determine that it unambiguously defines the meaning of all implementation-dependent options.

1193 The documentation of software development tools should include definitions of implementation-dependent options that may affect the meaning of the executable code, and those that are different from the standard language as documented. Where source code is provided to the evaluator, information should also be provided on compilation and linking options used.

1194 The documentation for hardware design and development tools should describe the use of all options that affect the output from the tools (e.g. detailed hardware specifications, or actual hardware).

13.8.2 Evaluation of sub-activity (ALC_TAT.2)

13.8.2.1 Objectives

1195 The objective of this sub-activity is to determine whether the developer has used well-defined development tools (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results, and whether implementation standards have been applied.

13.8.2.2 Input

1196 The evaluation evidence for this sub-activity is:

a) the development tool documentation;

b) the implementation standards description;

c) the provided implementation representation of the TSF.

13.8.2.3 Application notes

1197 This work may be performed in parallel with the evaluation activities under ADV_IMP, specifically with regard to determining the use of features in the tools that will affect the object code (e.g. compilation options).

13.8.2.4 Action ALC_TAT.2.1E

ALC_TAT.2.1C ***Each development tool used for implementation shall be well-defined.***

ALC_TAT.2-1 The evaluator ***shall examine*** the development tool documentation provided to determine that each development tool is well-defined.

1198 For example, a well-defined language, compiler or CAD system may be considered to be one that conforms to a recognised standard, such as the ISO standards. A well-defined language is one that has a clear and complete description of its syntax, and a detailed description of the semantics of each construct.

ALC_TAT.2.2C ***The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.***

ALC_TAT.2-2 The evaluator ***shall examine*** the documentation of each development tool to determine that it unambiguously defines the meaning of all statements as well as all conventions and directives used in the implementation.

1199 The development tool documentation (e.g. programming language specifications and user manuals) should cover all statements used in the

implementation representation of the TOE, and for each such statement should provide a clear and unambiguous definition of the purpose and effect of that statement. This work may be performed in parallel with the evaluator's examination of the implementation representation performed during the ADV_IMP sub-activity. The key test the evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to be able to understand the implementation representation. The documentation should not assume (for example) that the reader is an expert in the programming language used.

1200 Reference to the use of a documented standard is an acceptable approach to meet this requirement, provided that the standard is available to the evaluator. Any differences from the standard should be documented.

1201 The critical test is whether the evaluator can understand the TOE source code when performing source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can additionally be used in searching for problem areas:

a) In the language definition, phrases such as “the effect of this construct is undefined” and terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.

b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a common source of ambiguity problems.

c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is often poorly defined.

1202 Most languages in common use, however well designed, will have some problematic constructs. If the implementation language is mostly well defined, but some problematic constructs exist, then an inconclusive verdict should be assigned, pending examination of the source code.

1203 The evaluator should verify, during the examination of source code, that any use of the problematic constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs precluded by the documented standard are not used.

1204 The development tool documentation should define all conventions and directives used in the implementation.

ALC_TAT.2.3C ***The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.***

ALC_TAT.2-3 The evaluator ***shall examine*** the development tool documentation to determine that it unambiguously defines the meaning of all implementation-dependent options.

1205 The documentation of software development tools should include definitions of implementation-dependent options that may affect the meaning of the

executable code, and those that are different from the standard language as documented. Where source code is provided to the evaluator, information should also be provided on compilation and linking options used.

1206 The documentation for hardware design and development tools should describe the use of all options that affect the output from the tools (e.g. detailed hardware specifications, or actual hardware).

13.8.2.5 Action ALC_TAT.2.2E

ALC_TAT.2-4 The evaluator ***shall examine*** aspects of the implementation process to determine that documented implementation standards have been applied.

1207 This work unit requires the evaluator to analyse the provided implementation representation of the TOE to determine whether the documented implementation standards have been applied.

1208 The evaluator should verify that constructs excluded by the documented standard are not used.

1209 Additionally, the evaluator should verify the developer's procedures which ensure the application of the defined standards within the design and implementation process of the TOE. Therefore, documentary evidence should be supplemented by visiting the development environment. A visit to the development environment will allow the evaluator to:

a) observe the application of defined standards;

b) examine documentary evidence of application of procedures describing the use of defined standards;

c) interview development staff to check awareness of the application of defined standards and procedures.

1210 A development site visit is a useful means of gaining confidence in the procedures being used. Any decision not to make such a visit should be determined in consultation with the evaluation authority.

1211 The evaluator compares the provided implementation representation with the description of the applied implementation standards and verifies their use.

1212 At this level it is not required that the complete provided implementation representation of the TSF is based on implementation standards, but only those parts that are developed by the TOE developer himself. The evaluator may consult the configuration list required by the CM scope (ALC_CMS) to get the information which parts are developed by the TOE developer, and which by third party developers.

1213 If the referenced implementation standards are not applied for at least parts of the provided implementation representation, this work unit fails.

1214 Note that parts of the TOE which are not TSF relevant do not need to be examined.

1215 This work unit may be performed in conjunction with the evaluation activities under ADV_IMP.

13.8.3 Evaluation of sub-activity (ALC_TAT.3)

13.8.3.1 Objectives

1216 The objective of this sub-activity is to determine whether the developer and his subcontractors have used well-defined development tools (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results, and whether implementation standards have been applied.

13.8.3.2 Input

1217 The evaluation evidence for this sub-activity is:

a) the development tool documentation;

b) the implementation standards description;

c) the provided implementation representation of the TSF.

13.8.3.3 Application notes

1218 This work may be performed in parallel with the evaluation activities under ADV_IMP, specifically with regard to determining the use of features in the tools that will affect the object code (e.g. compilation options).

13.8.3.4 Action ALC_TAT.3.1E

ALC_TAT.3.1C ***Each development tool used for implementation shall be well-defined.***

ALC_TAT.3-1 The evaluator ***shall examine*** the development tool documentation provided to determine that each development tool is well-defined.

1219 For example, a well-defined language, compiler or CAD system may be considered to be one that conforms to a recognised standard, such as the ISO standards. A well-defined language is one that has a clear and complete description of its syntax, and a detailed description of the semantics of each construct.

1220 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

ALC_TAT.3.2C ***The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.***

ALC_TAT.3-2 The evaluator ***shall examine*** the documentation of each development tool to determine that it unambiguously defines the meaning of all statements as well as all conventions and directives used in the implementation.

1221 The development tool documentation (e.g. programming language specifications and user manuals) should cover all statements used in the implementation representation of the TOE, and for each such statement should provide a clear and unambiguous definition of the purpose and effect of that statement. This work may be performed in parallel with the evaluator's examination of the implementation representation performed during the ADV_IMP sub-activity. The key test the evaluator should apply is whether or not the documentation is sufficiently clear for the evaluator to be able to understand the implementation representation. The documentation should not assume (for example) that the reader is an expert in the programming language used.

1222 Reference to the use of a documented standard is an acceptable approach to meet this requirement, provided that the standard is available to the evaluator. Any differences from the standard should be documented.

1223 The critical test is whether the evaluator can understand the TOE source code when performing source code analysis covered in the ADV_IMP sub-activity. However, the following checklist can additionally be used in searching for problem areas:

a) In the language definition, phrases such as “the effect of this construct is undefined” and terms such as “implementation dependent” or “erroneous” may indicate ill-defined areas.

b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a common source of ambiguity problems.

c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is often poorly defined.

1224 Most languages in common use, however well designed, will have some problematic constructs. If the implementation language is mostly well defined, but some problematic constructs exist, then an inconclusive verdict should be assigned, pending examination of the source code.

1225 The evaluator should verify, during the examination of source code, that any use of the problematic constructs does not introduce vulnerabilities. The evaluator should also ensure that constructs precluded by the documented standard are not used.

1226 The development tool documentation should define all conventions and directives used in the implementation.

1227 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

ALC_TAT.3.3C ***The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.***

ALC_TAT.3-3 The evaluator ***shall examine*** the development tool documentation to determine that it unambiguously defines the meaning of all implementation-dependent options.

1228 The documentation of software development tools should include definitions of implementation-dependent options that may affect the meaning of the executable code, and those that are different from the standard language as documented. Where source code is provided to the evaluator, information should also be provided on compilation and linking options used.

1229 The documentation for hardware design and development tools should describe the use of all options that affect the output from the tools (e.g. detailed hardware specifications, or actual hardware).

1230 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

13.8.3.5 Action ALC_TAT.3.2E

ALC_TAT.3-4 The evaluator ***shall examine*** aspects of the implementation process to determine that documented implementation standards have been applied.

1231 This work unit requires the evaluator to analyse the provided implementation representation of the TOE to determine whether the documented implementation standards have been applied.

1232 The evaluator should verify that constructs excluded by the documented standard are not used.

1233 Additionally, the evaluator should verify the developer's procedures which ensure the application of the defined standards within the design and implementation process of the TOE. Therefore, documentary evidence should be supplemented by visiting the development environment. A visit to the development environment will allow the evaluator to:

a) observe the application of defined standards;

b) examine documentary evidence of application of procedures describing the use of defined standards;

c) interview development staff to check awareness of the application of defined standards and procedures.

1234 A development site visit is a useful means of gaining confidence in the procedures being used. Any decision not to make such a visit should be determined in consultation with the evaluation authority.

1235 The evaluator compares the provided implementation representation with the description of the applied implementation standards and verifies their use.

1236 At this level it is required that the complete provided implementation representation of the TSF is based on implementation standards, including third party contributions. This may require the evaluator to visit the sites of contributors. The evaluator may consult the configuration list required by the CM scope (ALC_CMS) to see who has developed which part of the TOE.

1237 Note that parts of the TOE which are not TSF relevant do not need to be examined.

1238 This work unit may be performed in conjunction with the evaluation activities under ADV_IMP.

14 Class ATE: Tests

14.1 Introduction

1239 The goal of this activity is to determine whether the TOE behaves as described in the ST and as specified in the evaluation evidence (described in the ADV class). This determination is achieved through some combination of the developer's own functional testing of the TSF (Functional tests (ATE_FUN)) and independent testing the TSF by the evaluator (Independent testing (ATE_IND)). At the lowest level of assurance, there is no requirement for developer involvement, so the only testing is conducted by the evaluator, using the limited available information about the TOE. Additional assurance is gained as the developer becomes increasingly involved both in testing and in providing additional information about the TOE, and as the evaluator increases the independent testing activities.

14.2 Application notes

1240 Testing of the TSF is conducted by the evaluator and, in most cases, by the developer. The evaluator's testing efforts consist not only of creating and running original tests, but also of assessing the adequacy of the developer's tests and re-running a subset of them.

1241 The evaluator analyses the developer's tests to determine the extent to which they are sufficient to demonstrate that TSFI (see Functional specification (ADV_FSP)) perform as specified, and to understand the developer's approach to testing. Similarly, the evaluator analyses the developer's tests to determine the extent to which they are sufficient to demonstrate the internal behaviour and properties of the TSF.

1242 The evaluator also executes a subset of the developer's tests as documented to gain confidence in the developer's test results: the evaluator will use the results of this analysis as an input to independently testing a subset of the TSF. With respect to this subset, the evaluator takes a testing approach that is different from that of the developer, particularly if the developer's tests have shortcomings.

1243 To determine the adequacy of developer's test documentation or to create new tests, the evaluator needs to understand the desired expected behaviour of the TSF, both internally and as seen at the TSFI, in the context of the SFRs it is to satisfy. The evaluator may choose to divide the TSF and TSFI into subsets according to functional areas of the ST (audit subsystem, audit-related TSFI, authentication module, authentication-related TSFI, etc.) if they were not already divided in the ST, and focus on one subset of the at a time, examining the ST requirement and the relevant parts of the development and guidance documentation to gain an understanding of the way the TOE is expected to behave. This reliance upon the development documentation underscores the need for the dependencies on ADV by Coverage (ATE_COV) and Depth (ATE_DPT).

1244 The CC has separated coverage and depth from functional tests to increase the flexibility when applying the components of the families. However, the requirements of the families are intended to be applied together to confirm that the TSF operates according to its specification. This tight coupling of families has led to some duplication of evaluator work units across sub-activities. These application notes are used to minimise duplication of text between sub-activities.

14.2.1 Understanding the expected behaviour of the TOE

1245 Before the adequacy of test documentation can be accurately evaluated, or before new tests can be created, the evaluator has to understand the desired expected behaviour of a security function in the context of the requirements it is to satisfy.

1246 As mentioned earlier, the evaluator may choose to subset the TSF and TSFI according to SFRs (audit, authentication, etc.) in the ST and focus on one subset at a time. The evaluator examines each ST requirement and the relevant parts of the functional specification and guidance documentation to gain an understanding of the way the related TSFI is expected to behave. Similarly, the evaluator examines the relevant parts of the TOE design and security architecture documentation to gain an understanding of the way the related modules or subsystems of the TSF are expected to behave.

1247 With an understanding of the expected behaviour, the evaluator examines the test plan to gain an understanding of the testing approach. In most cases, the testing approach will entail a TSFI being stimulated and its responses observed. Externally-visible functionality can be tested directly; however, in cases where functionality is not visible external to the TOE (for example, testing the residual information protection functionality), other means will need to be employed.

14.2.2 Testing vs. alternate approaches to verify the expected behaviour of functionality

1248 In cases where it is impractical or inadequate to test specific functionality (where it provides no externally-visible TSFI), the test plan should identify the alternate approach to verify expected behaviour. It is the evaluator's responsibility to determine the suitability of the alternate approach. However, the following should be considered when assessing the suitability of alternate approaches:

a) an analysis of the implementation representation to determine that the required behaviour should be exhibited by the TOE is an acceptable alternate approach. This could mean a code inspection for a software TOE or perhaps a chip mask inspection for a hardware TOE.

b) it is acceptable to use evidence of developer integration or module testing, even if the claimed assurance requirements do not include availability of lower level descriptions of the TOE modules (e.g. Evaluation of sub-activity (ADV_TDS.3)) or implementation

(Implementation representation (ADV_IMP)). If evidence of developer integration or module testing is used in verifying the expected behaviour of a security functionality, care should be given to confirm that the testing evidence reflects the current implementation of the TOE. If the subsystems or modules have been changed since testing occurred, evidence that the changes were tracked and addressed by analysis or further testing will usually be required.

1249 It should be emphasised that supplementing the testing effort with alternate approaches should only be undertaken when both the developer and evaluator determine that there exists no other practical means to test the expected behaviour.

14.2.3 Verifying the adequacy of tests

1250 Test pre-requisites are necessary to establish the required initial conditions for the test. They may be expressed in terms of parameters that must be set or in terms of test ordering in cases where the completion of one test establishes the necessary pre-requisites for another test. The evaluator must determine that the pre-requisites are complete and appropriate in that they will not bias the observed test results towards the expected test results.

1251 The test steps and expected results specify the actions and parameters to be applied to the TSFI as well as how the expected results should be verified and what they are. The evaluator must determine that the test steps and expected results are consistent with the descriptions of the TSFI in the functional specification. This means that each characteristic of the TSFI behaviour explicitly described in the functional specification should have tests and expected results to verify that behaviour.

1252 The overall aim of this testing activity is to determine that each subsystem, module, and TSFI has been sufficiently tested against the behavioural claims in the functional specification, TOE design, and architecture description. At the higher assurance levels, testing also includes bounds testing and negative testing. The test procedures will provide insight as to how the TSFIs, modules, and subsystems have been exercised by the developer during testing. The evaluator uses this information when developing additional tests to independently test the TSF.

14.3 Coverage (ATE_COV)

14.3.1 Evaluation of sub-activity (ATE_COV.1)

14.3.1.1 Objectives

1253 The objective of this sub-activity is to determine whether the developer has tested the TSFIs, 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.

14.3.1.2 Input

1254 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the test documentation;

d) the test coverage evidence.

14.3.1.3 Application notes

1255 The coverage analysis provided by the developer is required to show the correspondence between the tests provided as evaluation evidence and the functional specification. However, the coverage analysis need not demonstrate that all TSFI have been tested, or that all externally-visible interfaces to the TOE have been tested. Such shortcomings are considered by the evaluator during the independent testing (Evaluation of sub-activity (ATE_IND.2)) sub-activity.

14.3.1.4 Action ATE_COV.1.1E

ATE_COV.1.1C ***The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.***

ATE_COV.1-1 The evaluator ***shall examine*** the test coverage evidence to determine that the correspondence between the tests identified in the test documentation and the TSFIs described in the functional specification is accurate.

1256 Correspondence may take the form of a table or matrix. The coverage evidence required for this component will reveal the extent of coverage, rather than to show complete coverage. In cases where coverage is shown to be poor the evaluator should increase the level of independent testing to compensate.

14.3.2 Evaluation of sub-activity (ATE_COV.2)

14.3.2.1 Objectives

1257 The objective of this sub-activity is to determine whether the developer has tested all of the TSFIs, 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.

14.3.2.2 Input

a) the ST;

b) the functional specification;

c) the test documentation;

d) the test coverage analysis.

14.3.2.3 Action ATE_COV.2.1E

ATE_COV.2.1C ***The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.***

ATE_COV.2-1 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.

1258 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.

1259 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the functional specification.

ATE_COV.2-2 The evaluator ***shall examine*** the test plan to determine that the testing approach for each interface demonstrates the expected behaviour of that interface.

1260 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

ATE_COV.2-3 The evaluator ***shall examine*** the test procedures to determine that the test prerequisites, test steps and expected result(s) adequately test each interface.

1261 Guidance on this work units, as it pertains to the functional specification, can be found in:

a) 14.2.3, Verifying the adequacy of tests

ATE_COV.2.2C ***The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.***

ATE_COV.2-4 The evaluator ***shall examine*** the test coverage analysis to determine that the correspondence between the interfaces in the functional specification and the tests in the test documentation is complete.

1262 All TSFIs that are described in the functional specification have to be present in the test coverage analysis and mapped to tests in order for completeness to be claimed, although exhaustive specification testing of interfaces is not required. Incomplete coverage would be evident if an interface was identified in the functional specification and no test was mapped to it.

1263 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the functional specification.

14.3.3 Evaluation of sub-activity (ATE_COV.3)

1264 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

14.4 Depth (ATE_DPT)

14.4.1 Evaluation of sub-activity (ATE_DPT.1)

14.4.1.1 Objectives

1265 The objective of this sub-activity is to determine whether the developer has tested the TSF subsystems against the TOE design and the security architecture description.

14.4.1.2 Input

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the test documentation;

f) the depth of testing analysis.

14.4.1.3 Action ATE_DPT.1.1E

ATE_DPT.1.1C ***The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.***

ATE_DPT.1-1 The evaluator ***shall examine*** the depth of testing analysis to determine that the descriptions of the behaviour of TSF subsystems and of their interactions is included within the test documentation.

1266 This work unit verifies the content of the correspondence between the tests and the descriptions in the TOE design. In cases where the description of the TSF's architectural soundness (in Security Architecture (ADV_ARC)) cites specific mechanisms, this work unit also verifies the correspondence between the tests and the descriptions of the behaviour of such mechanisms.

1267 A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

1268 The evaluator is reminded that not all tests in the test documentation must map to a subsystem behaviour or interaction description.

ATE_DPT.1-2 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the behaviour of that subsystem as described in the TOE design.

1269 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1270 If TSF subsystem interfaces are described, the behaviour of those subsystems may be tested directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the behaviour that is described in the TOE design.

ATE_DPT.1-3 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the interactions among subsystems as described in the TOE design.

1271 While the previous work unit addresses behaviour of subsystems, this work unit addresses the interactions among work units.

1272 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1273 If TSF subsystem interfaces are described, the interactions with other subsystems may be tested directly from those interfaces. Otherwise, the interactions among subsystems must be inferred from the TSFI interfaces. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the interactions among subsystems that are described in the TOE design.

ATE_DPT.1.2C ***The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.***

ATE_DPT.1-4 The evaluator ***shall examine*** the test procedures to determine that all descriptions of TSF subsystem behaviour and interaction are tested.

1274 This work unit verifies the completeness of work unit ATE_DPT.1-1. All descriptions of TSF subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE design have to be tested. Incomplete depth of testing would be evident if a description of TSF subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design and no tests could be attributed to it.

1275 The evaluator is reminded that this does not imply that all tests in the test documentation must map to component interfaces in the TOE design.

14.4.2 Evaluation of sub-activity (ATE_DPT.2)

14.4.2.1 Objectives

1276 The objective of this sub-activity is to determine whether the developer has tested the TSF subsystems against the TOE design and the security architecture description.

14.4.2.2 Input

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the test documentation;

f) the depth of testing analysis.

14.4.2.3 Action ATE_DPT.2.1E

ATE_DPT.2.1C ***The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design.***

ATE_DPT.2-1 The evaluator ***shall examine*** the depth of testing analysis to determine that descriptions of the behaviour of TSF subsystems and of their interactions are included within the test documentation.

1277 This work unit verifies the content of the correspondence between the tests and the descriptions in the TOE design. In cases where the description of the TSF's architectural soundness (in Security Architecture (ADV_ARC)) cites specific mechanisms, this work unit also verifies the correspondence between the tests and the descriptions of the behaviour of such mechanisms.

1278 A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

1279 The evaluator is reminded that not all tests in the test documentation must map to a subsystem behaviour or interaction description.

ATE_DPT.2-2 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the behaviour of that subsystem as described in the TOE design.

1280 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1281 If TSF subsystem interfaces are described, the behaviour of those subsystems may be tested directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the behaviour that is described in the TOE design.

ATE_DPT.2-3 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the interactions among subsystems as described in the TOE design.

1282 While the previous work unit addresses behaviour of subsystems, this work unit addresses the interactions among work units.

1283 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1284 If TSF subsystem interfaces are described, the interactions with other subsystems may be tested directly from those interfaces. Otherwise, the interactions among subsystems must be inferred from the TSFI interfaces. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the interactions among subsystems that are described in the TOE design.

ATE_DPT.2-4 The evaluator ***shall examine*** the depth of testing analysis to determine that the interfaces of SFR-enforcing modules are included within the test documentation.

1285 This work unit verifies the content of the correspondence between the tests and the descriptions in the TOE design. In cases where the description of the TSF's architectural soundness (in Security Architecture (ADV_ARC)) cites specific mechanisms at the modular level, this work unit also verifies the correspondence between the tests and the descriptions of the behaviour of such mechanisms.

1286 A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

1287 The evaluator is reminded that not all tests in the test documentation must map to a subsystem behaviour or interaction description.

ATE_DPT.2-5 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for each TSF module interface demonstrates the expected behaviour of that interface.

1288 While work unit ATE_DPT.2-1 addresses expected behaviour of subsystems, this work unit addresses expected behaviour of the TSF module interfaces that are covered by ATE_DPT.2-4.

1289 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1290 Testing of an interface may be performed directly at that interface, or at the external interfaces, or a combination of both. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the interfaces. Specifically the evaluator determines whether testing at the internal interfaces is necessary or whether these internal interfaces can be adequately tested (albeit implicitly) by exercising the external interfaces. This determination is left to the evaluator, as is its justification.

ATE_DPT.2.2C ***The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.***

ATE_DPT.2-6 The evaluator ***shall examine*** the test procedures to determine that all descriptions of TSF subsystem behaviour and interaction are tested.

1291 This work unit verifies the completeness of work unit ATE_DPT.2-1. All descriptions of TSF subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE design have to be tested. Incomplete depth of testing would be evident if a description of TSF subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design and no tests could be attributed to it.

1292 The evaluator is reminded that this does not imply that all tests in the test documentation must map to component interfaces in the TOE design.

ATE_DPT.2.3C ***The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.***

ATE_DPT.2-7 The evaluator ***shall examine*** the test procedures to determine that all interfaces of SFR-enforcing modules are tested.

1293 This work unit verifies the completeness of work unit ATE_DPT.2-4. All interfaces of SFR-enforcing modules that are provided in the TOE design have to be tested. Incomplete depth of testing would be evident if any interface of any SFR-enforcing modules was identified in the TOE design and no tests could be attributed to it.

1294 The evaluator is reminded that this does not imply that all tests in the test documentation must map to an interface of an SFR-enforcing module in the TOE design.

14.4.3 Evaluation of sub-activity (ATE_DPT.3)

14.4.3.1 Objectives

1295 The objective of this sub-activity is to determine whether the developer has tested the TSF subsystems against the TOE design and the security architecture description.

14.4.3.2 Input

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the test documentation;

f) the depth of testing analysis.

14.4.3.3 Action ATE_DPT.3.1E

ATE_DPT.3.1C ***The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design.***

ATE_DPT.3-1 The evaluator ***shall examine*** the depth of testing analysis to determine that descriptions of the behaviour of TSF subsystems and of their interactions are included within the test documentation.

1296 This work unit verifies the content of the correspondence between the tests and the descriptions in the TOE design. A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

1297 The evaluator is reminded that not all tests in the test documentation must map to a subsystem behaviour or interaction description.

ATE_DPT.3-2 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the behaviour of that subsystem as described in the TOE design.

1298 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1299 If TSF subsystem interfaces are provided, the behaviour of those subsystems may be performed directly from those interfaces. Otherwise, the behaviour of those subsystems is tested from the TSFI interfaces. Or a combination of the two may be employed. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the behaviour that is described in the TOE design.

ATE_DPT.3-3 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for the behaviour description demonstrates the interactions among subsystems as described in the TOE design.

1300 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1301 If TSF subsystem interfaces are provided, the interactions with other subsystems may be performed directly from those interfaces. Otherwise, the interactions among subsystems must be inferred from the TSFI interfaces. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the interactions among subsystems that are described in the TOE design.

ATE_DPT.3-4 The evaluator ***shall examine*** the depth of testing analysis to determine that the interfaces of TSF modules are included within the test documentation.

1302 This work unit verifies the content of the correspondence between the tests and the descriptions in the TOE design. A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the behaviour/interaction presented in the depth-of coverage analysis has to be unambiguous.

1303 The evaluator is reminded that not all tests in the test documentation must map to a subsystem behaviour or interaction description.

ATE_DPT.3-5 The evaluator ***shall examine*** the test plan, test prerequisites, test steps and expected result(s) to determine that the testing approach for each TSF module interface demonstrates the expected behaviour of that interface.

1304 Guidance on this work unit can be found in:

a) 14.2.1, Understanding the expected behaviour of the TOE

b) 14.2.2, Testing vs. alternate approaches to verify the expected behaviour of functionality

1305 Testing of an interface may be performed directly at that interface, or at the external interfaces, or a combination of both. Whatever strategy is used the evaluator will consider its appropriateness for adequately testing the interfaces. Specifically the evaluator determines whether testing at the internal interfaces is necessary or whether these internal interfaces can be adequately tested (albeit implicitly) by exercising the external interfaces. This determination is left to the evaluator, as is its justification.

ATE_DPT.3.2C ***The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.***

ATE_DPT.3-6 The evaluator ***shall examine*** the test procedures to determine that all descriptions of TSF subsystem behaviour and interaction are tested.

1306 This work unit verifies the completeness of work unit ATE_DPT.3-1. All descriptions of TSF subsystem behaviour and of interactions among TSF subsystems that are provided in the TOE design have to be tested. Incomplete depth of testing would be evident if a description of TSF subsystem behaviour or of interactions among TSF subsystems was identified in the TOE design and no tests could be attributed to it.

1307 The evaluator is reminded that this does not imply that all tests in the test documentation must map to component interfaces in the TOE design.

ATE_DPT.3.3C ***The analysis of the depth of testing shall demonstrate that all modules in the TOE design have been tested.***

ATE_DPT.3-7 The evaluator ***shall examine*** the test procedures to determine that all interfaces of all modules are tested.

1308 This work unit verifies the completeness of work unit ATE_DPT.3-4. All interfaces of TSF modules that are provided in the TOE design have to be tested. Incomplete depth of testing would be evident if any interface of any TSF module was identified in the TOE design and no tests could be attributed to it.

1309 The evaluator is reminded that this does not imply that all tests in the test documentation must map to an interface of a TSF module in the TOE design.

14.4.4 Evaluation of sub-activity (ATE_DPT.4)

1310 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

14.5 Functional tests (ATE_FUN)

14.5.1 Evaluation of sub-activity (ATE_FUN.1)

14.5.1.1 Objectives

1311 The objective of this sub-activity is to determine whether the developer correctly performed and documented the tests in the test documentation.

14.5.1.2 Input

1312 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the test documentation.

14.5.1.3 Application notes

1313 The extent to which the test documentation is required to cover the TSF is dependent upon the coverage assurance component.

1314 For the developer tests provided, the evaluator determines whether the tests are repeatable, and the extent to which the developer's tests can be used for the evaluator's independent testing effort. Any TSFI for which the developer's test results indicate that it might not perform as specified should be tested independently by the evaluator to determine whether or not it does.

14.5.1.4 Action ATE_FUN.1.1E

ATE_FUN.1.1C ***The test documentation shall consist of test plans, expected test results and actual test results.***

ATE_FUN.1-1 The evaluator ***shall check*** that the test documentation includes test plans, expected test results and actual test results.

1315 The evaluator checks that test plans, expected tests results and actual test results are included in the test documentation.

ATE_FUN.1.2C ***The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.***

ATE_FUN.1-2 The evaluator ***shall examine*** the test plan to determine that it describes the scenarios for performing each test.

1316 The evaluator determines that the test plan provides information about the test configuration being used: both on the configuration of the TOE and on any test equipment being used. This information should be detailed enough to ensure that the test configuration is reproducible.

1317 The evaluator also determines that the test plan provides information about how to execute the test: any necessary automated set-up procedures (and whether they require privilege to run), inputs to be applied, how these inputs are applied, how output is obtained, any automated clean-up procedures (and whether they require privilege to run), etc. This information should be detailed enough to ensure that the test is reproducible.

1318 The evaluator may wish to employ a sampling strategy when performing this work unit.

ATE_FUN.1-3 The evaluator ***shall examine*** the test plan to determine that the TOE test configuration is consistent with the ST.

1319 The TOE referred to in the developer's test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1320 It is possible for the ST to specify more than one configuration for evaluation. The evaluator verifies that all test configurations identified in the developer test documentation are consistent with the ST. For example, the ST might define configuration options that must be set, which could have an impact upon what constitutes the TOE by including or excluding additional portions. The evaluator verifies that all such variations of the TOE are considered.

1321 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1322 The evaluator may wish to employ a sampling strategy when performing this work unit.

1323 If this work unit is applied to a component TOE that might be used/integrated in a composed TOE (see Class ACO: Composition), the following will apply. In the instances that the component TOE under evaluation depends on other components in the operational environment to support their operation, the developer may wish to consider using the other component(s) that will be used in the composed TOE to fulfil the requirements of the operational environment as one of the test configurations. This will reduce the amount an additional testing that will be required for the composed TOE evaluation.

ATE_FUN.1-4 The evaluator ***shall examine*** the test plans to determine that sufficient instructions are provided for any ordering dependencies.

1324 Some steps may have to be performed to establish initial conditions. For example, user accounts need to be added before they can be deleted. An example of ordering dependencies on the results of other tests is the need to

perform actions in a test that will result in the generation of audit records, before performing a test to consider the searching and sorting of those audit records. Another example of an ordering dependency would be where one test case generates a file of data to be used as input for another test case.

1325 The evaluator may wish to employ a sampling strategy when performing this work unit.

ATE_FUN.1.3C ***The expected test results shall show the anticipated outputs from a successful execution of the tests.***

ATE_FUN.1-5 The evaluator ***shall examine*** the test documentation to determine that all expected tests results are included.

1326 The expected test results are needed to determine whether or not a test has been successfully performed. Expected test results are sufficient if they are unambiguous and consistent with expected behaviour given the testing approach.

1327 The evaluator may wish to employ a sampling strategy when performing this work unit.

ATE_FUN.1.4C ***The actual test results shall be consistent with the expected test results.***

ATE_FUN.1-6 The evaluator ***shall check*** that the actual test results in the test documentation are consistent with the expected test results in the test documentation.

1328 A comparison of the actual and expected test results provided by the developer will reveal any inconsistencies between the results. It may be that a direct comparison of actual results cannot be made until some data reduction or synthesis has been first performed. In such cases, the developer's test documentation should describe the process to reduce or synthesise the actual data.

1329 For example, the developer may need to test the contents of a message buffer after a network connection has occurred to determine the contents of the buffer. The message buffer will contain a binary number. This binary number would have to be converted to another form of data representation in order to make the test more meaningful. The conversion of this binary representation of data into a higher-level representation will have to be described by the developer in enough detail to allow an evaluator to perform the conversion process (i.e. synchronous or asynchronous transmission, number of stop bits, parity, etc.).

1330 It should be noted that the description of the process used to reduce or synthesise the actual data is used by the evaluator not to actually perform the necessary modification but to assess whether this process is correct. It is up to the developer to transform the expected test results into a format that allows an easy comparison with the actual test results.

1331 The evaluator may wish to employ a sampling strategy when performing this work unit.

ATE_FUN.1-7 The evaluator ***shall report*** the developer testing effort, outlining the testing approach, configuration, depth and results.

1332 The developer testing information recorded in the ETR allows the evaluator to convey the overall testing approach and effort expended on the testing of the TOE by the developer. The intent of providing this information is to give a meaningful overview of the developer testing effort. It is not intended that the information regarding developer testing in the ETR be an exact reproduction of specific test steps or results of individual tests. The intention is to provide enough detail to allow other evaluators and overseers to gain some insight about the developer's testing approach, amount of testing performed, TOE test configurations, and the overall results of the developer testing.

1333 Information that would typically be found in the ETR section regarding the developer testing effort is:

a) TOE test configurations. The particular configurations of the TOE that were tested, including whether any privileged code was required to set up the test or clean up afterwards;

b) testing approach. An account of the overall developer testing strategy employed;

c) testing results. A description of the overall developer testing results.

1334 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the developer testing effort.

14.5.2 Evaluation of sub-activity (ATE_FUN.2)

1335 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

14.6 Independent testing (ATE_IND)

14.6.1 Evaluation of sub-activity (ATE_IND.1)

14.6.1.1 Objectives

1336 The goal of this activity is to determine, by independently testing a subset of the TSFI, whether the TOE behaves as specified in the functional specification and guidance documentation.

14.6.1.2 Input

1337 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the operational user guidance;

d) the preparative user guidance;

e) the TOE suitable for testing.

14.6.1.3 Action ATE_IND.1.1E

ATE_IND.1.1C ***The TOE shall be suitable for testing.***

ATE_IND.1-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1338 The TOE provided by the developer should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1339 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1340 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1341 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

ATE_IND.1-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state.

1342 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1343 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

14.6.1.4 Action ATE_IND.1.2E

ATE_IND.1-3 The evaluator ***shall devise*** a test subset.

1344 The evaluator selects a test subset and testing strategy that is appropriate for the TOE. One extreme testing strategy would be to have the test subset contain as many interfaces as possible tested with little rigour. Another testing strategy would be to have the test subset contain a few interfaces based on their perceived relevance and rigorously test these interfaces.

1345 Typically the testing approach taken by the evaluator should fall somewhere between these two extremes. The evaluator should exercise most of the interfaces using at least one test, but testing need not demonstrate exhaustive specification testing.

1346 The evaluator, when selecting the subset of the interfaces to be tested, should consider the following factors:

a) The number of interfaces from which to draw upon for the test subset. Where the TSF includes only a small number of relatively simple interfaces, it may be practical to rigorously test all of the interfaces. In other cases this may not be cost-effective, and sampling is required.

b) Maintaining a balance of evaluation activities. The evaluator effort expended on the test activity should be commensurate with that expended on any other evaluation activity.

1347 The evaluator selects the interfaces to compose the subset. This selection will depend on a number of factors, and consideration of these factors may also influence the choice of test subset size:

a) Significance of interfaces. Those interfaces more significant than others should be included in the test subset. One major factor of “significance” is the security-relevance (SFR-enforcing interfaces would be more significant than SFR-supporting interfaces, which are

more significant than SFR-non-interfering interfaces; see CC Part 3 Section Functional specification (ADV_FSP)). The other major factor of "significance" is the number of SFRs mapping to this interface (as determined when identifying the correspondence between levels of abstraction in ADV).

b) Complexity of the interface. Complex interfaces may require complex tests that impose onerous requirements on the developer or evaluator, which may not be conducive to cost-effective evaluations. Conversely, they are a likely area to find errors and are good candidates for the subset. The evaluator will need to strike a balance between these considerations.

c) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and their inclusion in the subset may maximise the number of interfaces tested (albeit implicitly). Certain interfaces will typically be used to provide a variety of security functionality, and will tend to be the target of an effective testing approach.

d) Types of interfaces (e.g. programmatic, command-line, protocol). The evaluator should consider including tests for all different types of interfaces that the TOE supports.

e) Interfaces that give rise to features that are innovative or unusual. Where the TOE contains innovative or unusual features, which may feature strongly in marketing literature and guidance documents, the corresponding interfaces should be strong candidates for testing.

1348 This guidance articulates factors to consider during the selection process of an appropriate test subset, but these are by no means exhaustive.

ATE_IND.1-4 The evaluator ***shall produce*** test documentation for the test subset that is sufficiently detailed to enable the tests to be reproducible.

1349 With an understanding of the expected behaviour of the TSF, from the ST and the functional specification, the evaluator has to determine the most feasible way to test the interface. Specifically the evaluator considers:

a) the approach that will be used, for instance, whether an external interface will be tested, or an internal interface using a test harness, or will an alternate test approach be employed (e.g. in exceptional circumstances, a code inspection, if the implementation representation is available);

b) the interface(s) that will be used to test and observe responses;

c) the initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

d) special test equipment that will be required to either stimulate an interface (e.g. packet generators) or make observations of an interface (e.g. network analysers).

1350 The evaluator may find it practical to test each interface using a series of test cases, where each test case will test a very specific aspect of expected behaviour.

1351 The evaluator's test documentation should specify the derivation of each test, tracing it back to the relevant interface(s).

ATE_IND.1-5 The evaluator ***shall conduct*** testing.

1352 The evaluator uses the test documentation developed as a basis for executing tests on the TOE. The test documentation is used as a basis for testing but this does not preclude the evaluator from performing additional ad hoc tests. The evaluator may devise new tests based on behaviour of the TOE discovered during testing. These new tests are recorded in the test documentation.

ATE_IND.1-6 The evaluator ***shall record*** the following information about the tests that compose the test subset:

a) identification of the interface behaviour to be tested;

b) instructions to connect and setup all required test equipment as required to conduct the test;

c) instructions to establish all prerequisite test conditions;

d) instructions to stimulate the interface;

e) instructions for observing the behaviour of the interface;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE;

h) actual test results.

1353 The level of detail should be such that another evaluator could repeat the tests and obtain an equivalent result. While some specific details of the test results may be different (e.g. time and date fields in an audit record) the overall result should be identical.

1354 There may be instances when it is unnecessary to provide all the information presented in this work unit (e.g. the actual test results of a test may not require any analysis before a comparison between the expected results can be

made). The determination to omit this information is left to the evaluator, as is the justification.

ATE_IND.1-7 The evaluator ***shall check*** that all actual test results are consistent with the expected test results.

1355 Any differences in the actual and expected test results may indicate that the TOE does not perform as specified or that the evaluator test documentation may be incorrect. Unexpected actual results may require corrective maintenance to the TOE or test documentation and perhaps require re-running of impacted tests and modifying the test sample size and composition. This determination is left to the evaluator, as is its justification.

ATE_IND.1-8 The evaluator ***shall report*** in the ETR the evaluator testing effort, outlining the testing approach, configuration, depth and results.

1356 The evaluator testing information reported in the ETR allows the evaluator to convey the overall testing approach and effort expended on the testing activity during the evaluation. The intent of providing this information is to give a meaningful overview of the testing effort. It is not intended that the information regarding testing in the ETR be an exact reproduction of specific test instructions or results of individual tests. The intention is to provide enough detail to allow other evaluators and overseers to gain some insight about the testing approach chosen, amount of testing performed, TOE test configurations, and the overall results of the testing activity.

1357 Information that would typically be found in the ETR section regarding the evaluator testing effort is:

a) TOE test configurations. The particular configurations of the TOE that were tested;

b) subset size chosen. The amount of interfaces that were tested during the evaluation and a justification for the size;

c) selection criteria for the interfaces that compose the subset. Brief statements about the factors considered when selecting interfaces for inclusion in the subset;

d) interfaces tested. A brief listing of the interfaces that merited inclusion in the subset;

e) verdict for the activity. The overall judgement on the results of testing during the evaluation.

1358 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the testing the evaluator performed during the evaluation.

14.6.2 Evaluation of sub-activity (ATE_IND.2)

14.6.2.1 Objectives

1359 The goal of this activity is to determine, by independently testing a subset of the TSF, whether the TOE behaves as specified in the design documentation, and to gain confidence in the developer's test results by performing a sample of the developer's tests.

14.6.2.2 Input

1360 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design description;

d) the operational user guidance;

e) the preparative user guidance;

f) the configuration management documentation;

g) the test documentation;

h) the TOE suitable for testing.

14.6.2.3 Action ATE_IND.2.1E

ATE_IND.2.1C ***The TOE shall be suitable for testing.***

ATE_IND.2-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1361 The TOE provided by the developer and identified in the test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1362 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1363 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1364 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

ATE_IND.2-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1365 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1366 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

ATE_IND.2.2C ***The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.***

ATE_IND.2-3 The evaluator ***shall examine*** the set of resources provided by the developer to determine that they are equivalent to the set of resources used by the developer to functionally test the TSF

1367 The set of resource used by the developer is documented in the developer test plan, as considered in the Functional tests (ATE_FUN) family. The resource set may include laboratory access and special test equipment, among others. Resources that are not identical to those used by the developer need to be equivalent in terms of any impact they may have on test results.

14.6.2.4 Action ATE_IND.2.2E

ATE_IND.2-4 The evaluator ***shall conduct*** testing using a sample of tests found in the developer test plan and procedures.

1368 The overall aim of this work unit is to perform a sufficient number of the developer tests to confirm the validity of the developer's test results. The evaluator has to decide on the size of the sample, and the developer tests that will compose the sample (see A.2).

1369 All the developer tests can be traced back to specific interfaces. Therefore, the factors to consider in the selection of the tests to compose the sample are similar to those listed for subset selection in work-unit ATE_IND.2-6. Additionally, the evaluator may wish to employ a random sampling method to select developer tests to include in the sample.

ATE_IND.2-5 The evaluator ***shall check*** that all the actual test results are consistent with the expected test results.

1370 Inconsistencies between the developer's expected test results and actual test results will compel the evaluator to resolve the discrepancies. Inconsistencies

encountered by the evaluator could be resolved by a valid explanation and resolution of the inconsistencies by the developer.

1371 If a satisfactory explanation or resolution can not be reached, the evaluator's confidence in the developer's test results may be lessened and it may be necessary for the evaluator to increase the sample size to the extent that the subset identified in work unit ATE_IND.2-4 is adequately tested: deficiencies with the developer's tests need to result in either corrective action to the developer's tests or in the production of new tests by the evaluator.

14.6.2.5 Action ATE_IND.2.3E

ATE_IND.2-6 The evaluator ***shall devise*** a test subset.

1372 The evaluator selects a test subset and testing strategy that is appropriate for the TOE. One extreme testing strategy would be to have the test subset contain as many interfaces as possible tested with little rigour. Another testing strategy would be to have the test subset contain a few interfaces based on their perceived relevance and rigorously test these interfaces.

1373 Typically the testing approach taken by the evaluator should fall somewhere between these two extremes. The evaluator should exercise most of the interfaces using at least one test, but testing need not demonstrate exhaustive specification testing.

1374 The evaluator, when selecting the subset of the interfaces to be tested, should consider the following factors:

a) The developer test evidence. The developer test evidence consists of: the test documentation, the available test coverage analysis, and the available depth of testing analysis. The developer test evidence will provide insight as to how the TSF has been exercised by the developer during testing. The evaluator applies this information when developing new tests to independently test the TOE. Specifically the evaluator should consider:

   1) augmentation of developer testing for interfaces. The evaluator may wish to perform more of the same type of tests by varying parameters to more rigorously test the interface.

   2) supplementation of developer testing strategy for interfaces. The evaluator may wish to vary the testing approach of a specific interface by testing it using another test strategy.

b) The number of interfaces from which to draw upon for the test subset. Where the TSF includes only a small number of relatively simple interfaces, it may be practical to rigorously test all of them. In other cases this may not be cost-effective, and sampling is required.

c) Maintaining a balance of evaluation activities. The evaluator effort expended on the test activity should be commensurate with that expended on any other evaluation activity.

1375 The evaluator selects the interfaces to compose the subset. This selection will depend on a number of factors, and consideration of these factors may also influence the choice of test subset size:

a) Rigour of developer testing of the interfaces. Those interfaces that the evaluator determines require additional testing should be included in the test subset.

b) Developer test results. If the results of developer tests cause the evaluator to doubt that an interface is not properly implemented, then the evaluator should include such interfaces in the test subset.

c) Significance of interfaces. Those interfaces more significant than others should be included in the test subset. One major factor of "significance" is the security-relevance (SFR-enforcing interfaces would be more significant than SFR-supporting interfaces, which are more significant than SFR-non-interfering interfaces; see CC Part 3 Section ADV_FSP). The other major factor of "significance" is the number of SFRs mapping to this interface (as determined when identifying the correspondence between levels of abstraction in ADV).

d) Complexity of interfaces. Interfaces that require complex implementation may require complex tests that impose onerous requirements on the developer or evaluator, which may not be conducive to cost-effective evaluations. Conversely, they are a likely area to find errors and are good candidates for the subset. The evaluator will need to strike a balance between these considerations.

e) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and their inclusion in the subset may maximise the number of interfaces tested (albeit implicitly). Certain interfaces will typically be used to provide a variety of security functionality, and will tend to be the target of an effective testing approach.

f) Types of interfaces (e.g. programmatic, command-line, protocol). The evaluator should consider including tests for all different types of interfaces that the TOE supports.

g) Interfaces that give rise to features that are innovative or unusual. Where the TOE contains innovative or unusual features, which may feature strongly in marketing literature and guidance documents, the corresponding interfaces should be strong candidates for testing.

1376 This guidance articulates factors to consider during the selection process of an appropriate test subset, but these are by no means exhaustive.

ATE_IND.2-7 The evaluator ***shall produce*** test documentation for the test subset that is sufficiently detailed to enable the tests to be reproducible.

1377 With an understanding of the expected behaviour of the TSF, from the ST, the functional specification, and the TOE design description, the evaluator has to determine the most feasible way to test the interface. Specifically the evaluator considers:

a) the approach that will be used, for instance, whether an external interface will be tested, or an internal interface using a test harness, or will an alternate test approach be employed (e.g. in exceptional circumstances, a code inspection);

b) the interface(s) that will be used to test and observe responses;

c) the initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

d) special test equipment that will be required to either stimulate an interface (e.g. packet generators) or make observations of an interface (e.g. network analysers).

1378 The evaluator may find it practical to test each interface using a series of test cases, where each test case will test a very specific aspect of expected behaviour of that interface.

1379 The evaluator's test documentation should specify the derivation of each test, tracing it back to the relevant interface(s).

ATE_IND.2-8 The evaluator ***shall conduct*** testing.

1380 The evaluator uses the test documentation developed as a basis for executing tests on the TOE. The test documentation is used as a basis for testing but this does not preclude the evaluator from performing additional ad hoc tests. The evaluator may devise new tests based on behaviour of the TOE discovered during testing. These new tests are recorded in the test documentation.

ATE_IND.2-9 The evaluator ***shall record*** the following information about the tests that compose the test subset:

a) identification of the interface behaviour to be tested;

b) instructions to connect and setup all required test equipment as required to conduct the test;

c) instructions to establish all prerequisite test conditions;

d) instructions to stimulate the interface;

e) instructions for observing the interface;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE;

h) actual test results.

1381 The level of detail should be such that another evaluator could repeat the tests and obtain an equivalent result. While some specific details of the test results may be different (e.g. time and date fields in an audit record) the overall result should be identical.

1382 There may be instances when it is unnecessary to provide all the information presented in this work unit (e.g. the actual test results of a test may not require any analysis before a comparison between the expected results can be made). The determination to omit this information is left to the evaluator, as is the justification.

ATE_IND.2-10 The evaluator ***shall check*** that all actual test results are consistent with the expected test results.

1383 Any differences in the actual and expected test results may indicate that the TOE does not perform as specified or that the evaluator test documentation may be incorrect. Unexpected actual results may require corrective maintenance to the TOE or test documentation and perhaps require re-running of impacted tests and modifying the test sample size and composition. This determination is left to the evaluator, as is its justification.

ATE_IND.2-11 The evaluator ***shall report*** in the ETR the evaluator testing effort, outlining the testing approach, configuration, depth and results.

1384 The evaluator testing information reported in the ETR allows the evaluator to convey the overall testing approach and effort expended on the testing activity during the evaluation. The intent of providing this information is to give a meaningful overview of the testing effort. It is not intended that the information regarding testing in the ETR be an exact reproduction of specific test instructions or results of individual tests. The intention is to provide enough detail to allow other evaluators and overseers to gain some insight about the testing approach chosen, amount of evaluator testing performed, amount of developer tests performed, TOE test configurations, and the overall results of the testing activity.

1385 Information that would typically be found in the ETR section regarding the evaluator testing effort is:

a) TOE test configurations. The particular configurations of the TOE that were tested.

b) subset size chosen. The amount of interfaces that were tested during the evaluation and a justification for the size.

c) selection criteria for the interfaces that compose the subset. Brief statements about the factors considered when selecting interfaces for inclusion in the subset.

d) Interfaces tested. A brief listing of the interfaces that merited inclusion in the subset.

e) developer tests performed. The amount of developer tests performed and a brief description of the criteria used to select the tests.

f) verdict for the activity. The overall judgement on the results of testing during the evaluation.

1386 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the testing the evaluator performed during the evaluation.

14.6.3 Evaluation of sub-activity (ATE_IND.3)

1387 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

15 Class AVA: Vulnerability assessment

15.1 Introduction

1388 The purpose of the vulnerability assessment activity is to determine the exploitability of flaws or weaknesses in the TOE in the operational environment. This determination is based upon analysis of the evaluation evidence and a search of publicly available material by the evaluator and is supported by evaluator penetration testing.

15.2 Vulnerability analysis (AVA_VAN)

15.2.1 Evaluation of sub-activity (AVA_VAN.1)

15.2.1.1 Objectives

1389 The objective of this sub-activity is to determine whether the TOE, in its operational environment, has easily identifiable exploitable vulnerabilities.

15.2.1.2 Input

1390 The evaluation evidence for this sub-activity is:

a) the ST;

b) the guidance documentation;

c) the TOE suitable for testing;

d) information publicly available to support the identification of potential vulnerabilities.

1391 Other input for this sub-activity is:

a) current information regarding potential vulnerabilities (e.g. from an evaluation authority).

15.2.1.3 Application notes

1392 The evaluator should consider performing additional tests as a result of potential vulnerabilities encountered during the conduct of other parts of the evaluation.

1393 The use of the term guidance in this sub-activity refers to the operational guidance and the preparative guidance.

1394 Potential vulnerabilities may be in information that is publicly available, or not, and may require skill to exploit, or not. These two aspects are related, but are distinct. It should not be assumed that, simply because a potential vulnerability is identifiable from information that is publicly available, it can be easily exploited.

15.2.1.4 Action AVA_VAN.1.1E

AVA_VAN.1.1C ***The TOE shall be suitable for testing.***

AVA_VAN.1-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1395 The TOE provided by the developer and identified in the test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1396 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1397 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1398 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

AVA_VAN.1-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1399 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1400 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

15.2.1.5 Action AVA_VAN.1.2E

AVA_VAN.1-3 The evaluator ***shall examine*** sources of information publicly available to identify potential vulnerabilities in the TOE.

1401 The evaluator examines the sources of information publicly available to support the identification of possible potential vulnerabilities in the TOE. There are many sources of publicly available information, which should be considered, e.g. mailing lists and security forums on the world wide web that report known vulnerabilities in specified technologies.

1402 The evaluator should not constrain their consideration of publicly available information to the above, but should consider any other relevant information available.

1403 While examining the evidence provided the evaluator will use the information in the public domain to further search for potential vulnerabilities. Where the evaluators have identified areas of concern, the evaluator should consider information publicly available that relate to those areas of concern.

1404 The availability of information that may be readily available to an attacker that helps to identify and facilitate attacks effectively operates to substantially enhance the attack potential of a given attacker. The accessibility of vulnerability information and sophisticated attack tools on the Internet makes it more likely that this information will be used in attempts to identify potential vulnerabilities in the TOE and exploit them. Modern search tools make such information easily available to the evaluator, and the determination of resistance to published potential vulnerabilities and well known generic attacks can be achieved in a cost-effective manner.

1405 The search of the information publicly available should be focused on those sources that refer specifically to the product from which the TOE is derived. The extensiveness of this search should consider the following factors: TOE type, evaluator experience in this TOE type, expected attack potential and the level of ADV evidence available.

1406 The identification process is iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1407 The evaluator will report what actions were taken to identify potential vulnerabilities in the information publicly available. However, in this type of search, the evaluator may not be able to describe the steps in identifying potential vulnerabilities before the outset of the examination, as the approach may evolve as a result of findings during the search.

1408 The evaluator will report the evidence examined in completing the search for potential vulnerabilities.

AVA_VAN.1-4 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.

1409 It may be identified that no further consideration of the potential vulnerability is required if for example the evaluator identifies that measures in the operational environment, either IT or non-IT, prevent exploitation of the potential vulnerability in that operational environment. For instance, restricting physical access to the TOE to authorised users only may effectively render a potential vulnerability to tampering unexploitable.

1410 The evaluator records any reasons for exclusion of potential vulnerabilities from further consideration if the evaluator determines that the potential vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1411 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15.2.1.6 Action AVA_VAN.1.3E

AVA_VAN.1-5 The evaluator ***shall devise*** penetration tests, based on the independent search for potential vulnerabilities.

1412 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the TOE, in its operational environment, to the potential vulnerabilities identified during the search of the sources of information publicly available. The evaluator should have access to current information (e.g. from the evaluation authority) regarding known potential vulnerabilities that may not have been considered by the evaluator, and may also have encountered potential vulnerabilities as a result of performing other evaluation activities.

1413 The evaluator will probably find it practical to carry out penetration test using a series of test cases, where each test case will test for a specific potential vulnerability.

1414 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers a potential vulnerability that is beyond Basic attack potential, this is reported in the ETR as a residual vulnerability.

AVA_VAN.1-6 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include:

a) identification of the potential vulnerability the TOE is being tested for;

b) instructions to connect and setup all required test equipment as required to conduct the penetration test;

c) instructions to establish all penetration test prerequisite initial conditions;

d) instructions to stimulate the TSF;

e) instructions for observing the behaviour of the TSF;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE.

1415 The evaluator prepares for penetration testing based on the list of potential vulnerabilities identified during the search of the public domain.

1416 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond those for which a Basic attack potential is required to effect an attack. However, as a result of evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only by an attacker with greater than Basic attack potential. Such vulnerabilities are to be reported in the ETR as residual vulnerabilities.

1417 With an understanding of the potential vulnerability, the evaluator determines the most feasible way to test for the TOE's susceptibility. Specifically the evaluator considers:

a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses;

b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

c) special test equipment that will be required to either stimulate a TSFI or make observations of a TSFI (although it is unlikely that specialist equipment would be required to exploit a potential vulnerability assuming a Basic attack potential);

d) whether theoretical analysis should replace physical testing, particularly relevant where the results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are likely to succeed after a given number of attempts.

1418 The evaluator will probably find it practical to carry out penetration testing using a series of test cases, where each test case will test for a specific potential vulnerability.

1419 The intent of specifying this level of detail in the test documentation is to allow another evaluator to repeat the tests and obtain an equivalent result.

AVA_VAN.1-7 The evaluator ***shall conduct*** penetration testing.

1420 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.1-5 as a basis for executing penetration tests on the TOE, but this does not preclude the evaluator from performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests as a result of information learnt during penetration testing that, if performed by the evaluator, are to be recorded in the penetration test documentation. Such tests may be required to follow up unexpected results or observations, or to investigate potential vulnerabilities suggested to the evaluator during the pre-planned testing.

1421 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation

expertise, the evaluator discovers a potential vulnerability that is beyond Basic attack potential, this is reported in the ETR as a residual vulnerability.

AVA_VAN.1-8 The evaluator ***shall record*** the actual results of the penetration tests.

1422 While some specific details of the actual test results may be different from those expected (e.g. time and date fields in an audit record) the overall result should be identical. Any unexpected test results should be investigated. The impact on the evaluation should be stated and justified.

AVA_VAN.1-9 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.

1423 The penetration testing information reported in the ETR allows the evaluator to convey the overall penetration testing approach and effort expended on this sub-activity. The intent of providing this information is to give a meaningful overview of the evaluator's penetration testing effort. It is not intended that the information regarding penetration testing in the ETR be an exact reproduction of specific test steps or results of individual penetration tests. The intention is to provide enough detail to allow other evaluators and evaluation authorities to gain some insight about the penetration testing approach chosen, amount of penetration testing performed, TOE test configurations, and the overall results of the penetration testing activity.

1424 Information that would typically be found in the ETR section regarding evaluator penetration testing efforts is:

a) TOE test configurations. The particular configurations of the TOE that were penetration tested;

b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of the penetration testing;

c) verdict for the sub-activity. The overall judgement on the results of penetration testing.

1425 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the penetration testing the evaluator performed during the evaluation.

AVA_VAN.1-10 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Basic attack potential.

1426 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by an attacker possessing less than Enhanced-Basic attack potential, then this evaluator action fails.

1427 The guidance in B.4 should be used to determine the attack potential required to exploit a particular vulnerability and whether it can therefore be exploited in the intended environment. It may not be necessary for the attack potential

to be calculated in every instance, only if there is some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an attack potential less than Enhanced-Basic.

AVA_VAN.1-11 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each:

a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);

b) the SFR(s) not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual).

e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.

15.2.2 Evaluation of sub-activity (AVA_VAN.2)

15.2.2.1 Objectives

1428 The objective of this sub-activity is to determine whether the TOE, in its operational environment, has vulnerabilities exploitable by attackers possessing Basic attack potential.

15.2.2.2 Input

1429 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the guidance documentation;

f) the TOE suitable for testing;

g) information publicly available to support the identification of possible potential vulnerabilities.

1430 The remaining implicit evaluation evidence for this sub-activity depends on the components that have been included in the assurance package. The evidence provided for each component is to be used as input in this sub-activity.

1431 Other input for this sub-activity is:

a) current information regarding public domain potential vulnerabilities and attacks (e.g. from an evaluation authority).

15.2.2.3 Application notes

1432 The evaluator should consider performing additional tests as a result of potential vulnerabilities encountered during other parts of the evaluation.

15.2.2.4 Action AVA_VAN.2.1E

AVA_VAN.2.1C ***The TOE shall be suitable for testing.***

AVA_VAN.2-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1433 The TOE provided by the developer and identified in the test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1434 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1435 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1436 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

AVA_VAN.2-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1437 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1438 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

15.2.2.5 Action AVA_VAN.2.2E

AVA_VAN.2-3 The evaluator ***shall examine*** sources of information publicly available to identify potential vulnerabilities in the TOE.

1439 The evaluator examines the sources of information publicly available to support the identification of possible potential vulnerabilities in the TOE. There are many sources of publicly available information which the evaluator should consider using items such as those available on the world wide web, including:

a) specialist publications (magazines, books);

b) research papers.

1440 The evaluator should not constrain their consideration of publicly available information to the above, but should consider any other relevant information available.

1441 While examining the evidence provided the evaluator will use the information in the public domain to further search for potential vulnerabilities. Where the evaluators have identified areas of concern, the evaluator should consider information publicly available that relate to those areas of concern.

1442 The availability of information that may be readily available to an attacker that helps to identify and facilitate attacks may substantially enhance the attack potential of a given attacker. The accessibility of vulnerability information and sophisticated attack tools on the Internet makes it more likely that this information will be used in attempts to identify potential vulnerabilities in the TOE and exploit them. Modern search tools make such information easily available to the evaluator, and the determination of resistance to published potential vulnerabilities and well known generic attacks can be achieved in a cost-effective manner.

1443 The search of the information publicly available should be focused on those sources that refer specifically to the product from which the TOE is derived. The extensiveness of this search should consider the following factors: TOE type, evaluator experience in this TOE type, expected attack potential and the level of ADV evidence available.

1444 The identification process is iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1445 The evaluator will report what actions were taken to identify potential vulnerabilities in the evidence. However, in this type of search, the evaluator may not be able to describe the steps in identifying potential vulnerabilities before the outset of the examination, as the approach may evolve as a result of findings during the search.

1446 The evaluator will report the evidence examined in completing the search for potential vulnerabilities. This selection of evidence may be derived from those areas of concern identified by the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to another rationale provided by the evaluator.

15.2.2.6 Action AVA_VAN.2.3E

AVA_VAN.2-4 The evaluator ***shall conduct*** a search of ST, guidance documentation, functional specification, TOE design and security architecture description evidence to identify possible potential vulnerabilities in the TOE.

1447 A search of the evidence should be completed whereby specifications and documentation for the TOE are analysed and then potential vulnerabilities in the TOE are hypothesised, or speculated. The list of hypothesised potential vulnerabilities is then prioritised on the basis of the estimated probability that a potential vulnerability exists and, assuming an exploitable vulnerability does exist the attack potential required to exploit it, and on the extent of control or compromise it would provide. The prioritised list of potential vulnerabilities is used to direct penetration testing against the TOE.

1448 The security architecture description provides the developer vulnerability analysis, as it documents how the TSF protects itself from interference from untrusted subjects and prevents the bypass of security enforcement functionality. Therefore, the evaluator should use this description of the protection of the TSF as a basis for the search for possible ways to undermine the TSF.

1449 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent vulnerability analysis should consider generic potential vulnerabilities under each of the following headings:

a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be supplied by the evaluation authority;

b) bypassing;

c) tampering;

d) direct attacks;

e) monitoring;

f) misuse.

1450 Items b) - f) are explained in greater detail in Annex B.

1451 The security architecture description should be considered in light of each of the above generic potential vulnerabilities. Each potential vulnerability should be considered to search for possible ways in which to defeat the TSF protection and undermine the TSF.

AVA_VAN.2-5 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.

1452 It may be identified that no further consideration of the potential vulnerability is required if for example the evaluator identifies that measures in the operational environment, either IT or non-IT, prevent exploitation of the potential vulnerability in that operational environment. For instance, restricting physical access to the TOE to authorised users only may effectively render a potential vulnerability to tampering unexploitable.

1453 The evaluator records any reasons for exclusion of potential vulnerabilities from further consideration if the evaluator determines that the potential vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1454 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15.2.2.7 Action AVA_VAN.2.4E

AVA_VAN.2-6 The evaluator ***shall devise*** penetration tests, based on the independent search for potential vulnerabilities.

1455 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the TOE, in its operational environment, to the potential vulnerabilities identified during the search of the sources of publicly available information and the analysis of the TOE guidance and design evidence. The evaluator should have access to current information (e.g. from the evaluation authority) regarding known potential vulnerabilities that may not have been considered by the evaluator.

1456 The evaluator is reminded that, as for considering the security architecture description in the search for vulnerabilities (as detailed in AVA_VAN.2-3), testing should be performed to confirm the architectural properties. This is likely to require negative tests attempting to disprove the properties of the security architecture. In developing the strategy for penetration testing, the evaluator will ensure that each of the major characteristics of the security architecture description are tested, either in functional testing (as considered in 14) or evaluator penetration testing.

1457 The evaluator will probably find it practical to carry out penetration test using a series of test cases, where each test case will test for a specific potential vulnerability.

1458 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation

expertise, the evaluator discovers an exploitable vulnerability that is beyond Basic attack potential, this is reported in the ETR as a residual vulnerability.

1459 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be found in Annex B.4.

1460 Potential vulnerabilities hypothesised as exploitable only by attackers possessing Enhanced-Basic, Moderate or High attack potential do not result in a failure of this evaluator action. Where analysis supports the hypothesis, these need not be considered further as an input to penetration testing. However, such vulnerabilities are reported in the ETR as residual vulnerabilities.

1461 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Basic attack potential and resulting in a violation of the security objectives should be the highest priority potential vulnerabilities comprising the list used to direct penetration testing against the TOE.

AVA_VAN.2-7 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include:

a) identification of the potential vulnerability the TOE is being tested for;

b) instructions to connect and setup all required test equipment as required to conduct the penetration test;

c) instructions to establish all penetration test prerequisite initial conditions;

d) instructions to stimulate the TSF;

e) instructions for observing the behaviour of the TSF;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE.

1462 The evaluator prepares for penetration testing based on the list of potential vulnerabilities identified during the search of the public domain and the analysis of the evaluation evidence.

1463 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond those for which a Basic attack potential is required to effect an attack. However, as a result of evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only by an attacker with greater than Basic attack potential. Such vulnerabilities are to be reported in the ETR as residual vulnerabilities.

1464 With an understanding of the potential vulnerability, the evaluator determines the most feasible way to test for the TOE's susceptibility. Specifically the evaluator considers:

a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses (It is possible that the evaluator will need to use an interface to the TOE other than the TSFI to demonstrate properties of the TSF such as those described in the security architecture description (as required by ADV_ARC). It should the noted, that although these TOE interfaces provide a means of testing the TSF properties, they are not the subject of the test.);

b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

c) special test equipment that will be required to either stimulate a TSFI or make observations of a TSFI (although it is unlikely that specialist equipment would be required to exploit a potential vulnerability assuming a Basic attack potential);

d) whether theoretical analysis should replace physical testing, particularly relevant where the results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are likely to succeed after a given number of attempts.

1465 The evaluator will probably find it practical to carry out penetration testing using a series of test cases, where each test case will test for a specific potential vulnerability.

1466 The intent of specifying this level of detail in the test documentation is to allow another evaluator to repeat the tests and obtain an equivalent result.

AVA_VAN.2-8 The evaluator ***shall conduct*** penetration testing.

1467 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.2-6 as a basis for executing penetration tests on the TOE, but this does not preclude the evaluator from performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests as a result of information learnt during penetration testing that, if performed by the evaluator, are to be recorded in the penetration test documentation. Such tests may be required to follow up unexpected results or observations, or to investigate potential vulnerabilities suggested to the evaluator during the pre-planned testing.

1468 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the evaluator should determine whether or not the evaluator's own analysis was incorrect, or if evaluation deliverables are incorrect or incomplete.

1469 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond basic attack potential, this is reported in the ETR as a residual vulnerability.

AVA_VAN.2-9 The evaluator ***shall record*** the actual results of the penetration tests.

1470 While some specific details of the actual test results may be different from those expected (e.g. time and date fields in an audit record) the overall result should be identical. Any unexpected test results should be investigated. The impact on the evaluation should be stated and justified.

AVA_VAN.2-10 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.

1471 The penetration testing information reported in the ETR allows the evaluator to convey the overall penetration testing approach and effort expended on this sub-activity. The intent of providing this information is to give a meaningful overview of the evaluator's penetration testing effort. It is not intended that the information regarding penetration testing in the ETR be an exact reproduction of specific test steps or results of individual penetration tests. The intention is to provide enough detail to allow other evaluators and evaluation authorities to gain some insight about the penetration testing approach chosen, amount of penetration testing performed, TOE test configurations, and the overall results of the penetration testing activity.

1472 Information that would typically be found in the ETR section regarding evaluator penetration testing efforts is:

a) TOE test configurations. The particular configurations of the TOE that were penetration tested;

b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of the penetration testing;

c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.

1473 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the penetration testing the evaluator performed during the evaluation.

AVA_VAN.2-11 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Basic attack potential.

1474 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by an attacker possessing less than an Enhanced-Basic attack potential, then this evaluator action fails.

1475 The guidance in B.4 should be used to determine the attack potential required to exploit a particular vulnerability and whether it can therefore be exploited in the intended environment. It may not be necessary for the attack potential to be calculated in every instance, only if there is some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an attack potential less than Enhanced-Basic.

AVA_VAN.2-12 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each:

a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);

b) the SFR(s) not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual).

e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.

15.2.3 Evaluation of sub-activity (AVA_VAN.3)

15.2.3.1 Objectives

1476 The objective of this sub-activity is to determine whether the TOE, in its operational environment, has vulnerabilities exploitable by attackers possessing Enhanced-Basic attack potential.

15.2.3.2 Input

1477 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the implementation subset selected;

f) the guidance documentation;

g) the TOE suitable for testing;

h) information publicly available to support the identification of possible potential vulnerabilities.

1478 The remaining implicit evaluation evidence for this sub-activity depends on the components that have been included in the assurance package. The evidence provided for each component is to be used as input in this sub-activity.

1479 Other input for this sub-activity is:

a) current information regarding public domain potential vulnerabilities and attacks (e.g. from an evaluation authority).

15.2.3.3 Application notes

1480 During the conduct of evaluation activities the evaluator may also identify areas of concern. These are specific portions of the TOE evidence that the evaluator has some reservation about, although the evidence meets the requirements for the activity with which the evidence is associated. For example, a particular interface specification looks particularly complex, and therefore may be prone to error either in the development of the TOE or in the operation of the TOE. There is no potential vulnerability apparent at this stage, further investigation is required. This is beyond the bounds of encountered, as further investigation is required.

1481 The focused approach to the identification of potential vulnerabilities is an analysis of the evidence with the aim of identifying any potential vulnerabilities evident through the contained information. It is an unstructured analysis, as the approach is not predetermined. Further guidance on focused vulnerability analysis can be found in Annex B.2.2.2.2.

15.2.3.4 Action AVA_VAN.3.1E

AVA_VAN.3.1C ***The TOE shall be suitable for testing.***

AVA_VAN.3-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1482 The TOE provided by the developer and identified in the test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1483 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1484 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some

objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1485 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

AVA_VAN.3-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1486 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1487 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

15.2.3.5 Action AVA_VAN.3.2E

AVA_VAN.3-3 The evaluator ***shall examine*** sources of information publicly available to identify potential vulnerabilities in the TOE.

1488 The evaluator examines the sources of information publicly available to support the identification of possible potential vulnerabilities in the TOE. There are many sources of publicly available information which the evaluator should consider using items such as those available on the world wide web, including:

a) specialist publications (magazines, books);

b) research papers;

c) conference proceedings.

1489 The evaluator should not constrain their consideration of publicly available information to the above, but should consider any other relevant information available.

1490 While examining the evidence provided the evaluator will use the information in the public domain to further search for potential vulnerabilities. Where the evaluators have identified areas of concern, the evaluator should consider information publicly available that relate to those areas of concern.

1491 The availability of information that may be readily available to an attacker that helps to identify and facilitate attacks may substantially enhance the

attack potential of a given attacker. The accessibility of vulnerability information and sophisticated attack tools on the Internet makes it more likely that this information will be used in attempts to identify potential vulnerabilities in the TOE and exploit them. Modern search tools make such information easily available to the evaluator, and the determination of resistance to published potential vulnerabilities and well known generic attacks can be achieved in a cost-effective manner.

1492 The search of the information publicly available should be focused on those sources that refer to the technologies used in the development of the product from which the TOE is derived. The extensiveness of this search should consider the following factors: TOE type, evaluator experience in this TOE type, expected attack potential and the level of ADV evidence available.

1493 The identification process is iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1494 The evaluator will report what actions were taken to identify potential vulnerabilities in the evidence. However, in this type of search, the evaluator may not be able to describe the steps in identifying potential vulnerabilities before the outset of the examination, as the approach may evolve as a result of findings during the search.

1495 The evaluator will report the evidence examined in completing the search for potential vulnerabilities. This selection of evidence may be derived from those areas of concern identified by the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to another rationale provided by the evaluator.

15.2.3.6 Action AVA_VAN.3.3E

AVA_VAN.3-4 The evaluator ***shall conduct*** a focused search of ST, guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify possible potential vulnerabilities in the TOE.

1496 A flaw hypothesis methodology should be used whereby specifications and development and guidance evidence are analysed and then potential vulnerabilities in the TOE are hypothesised, or speculated.

1497 The evaluator should use the knowledge of the TOE design and operation gained from the TOE deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE and potential errors in the specified method of operation of the TOE.

1498 The security architecture description provides the developer vulnerability analysis, as it documents how the TSF protects itself from interference from untrusted subjects and prevents the bypass of security enforcement functionality. Therefore, the evaluator should build upon the understanding

of the TSF protection gained from the analysis of this evidence and then develop this in the knowledge gained from other development (11 evidence.

1499 The following provide some examples of hypotheses that may be created when examining the evidence:

a) consideration of malformed input for interfaces available to an attacker at the external interfaces;

b) examination of a key security mechanism cited in the security architecture description, such as process separation, hypothesising internal buffer overflows that may lead to degradation of separation;

c) search to identify any objects created in the TOE implementation representation that are then not fully controlled by the TSF, and could be used by an attacker to undermine SFRs.

1500 The approach taken is directed by areas of concern identified during examination of the evidence during the conduct of evaluation activities and ensuring a representative sample of the development and guidance evidence provided for the evaluation is searched.

1501 For guidance on sampling see Annex A.2. This guidance should be considered when selecting the subset, giving reasons for:

a) the approach used in selection;

b) qualification that the evidence to be examined supports that approach.

1502 The areas of concern may relate to the sufficiency of specific protection features detailed in the security architecture description.

1503 The evidence to be considered during the vulnerability analysis may be linked to the evidence the attacker is assumed to be able to obtain. For example, the developer may protect the TOE design and implementation representations, so the only information assumed to be available to an attacker is the functional specification and guidance (available publicly available). So, although the objectives for assurance in the TOE ensure the TOE design and implementation representation requirements are met, these design representations may only be searched to further investigate areas of concerns.

1504 On the other hand, if the source is available publicly available it would be reasonable to assume that the attacker has access to the source and can use this in attempts to attack the TOE. Therefore, the source should be considered in the focused examination approach.

1505 The following indicates examples for the selection of the subset of evidence to be considered:

a) For an evaluation where all levels of design abstraction from functional specification to implementation representation are

provided, examination of information in the functional specification and the implementation representation may be selected, as the functional specification provides detail of interfaces available to an attacker, and the implementation representation incorporates the design decisions made at all other design abstractions. Therefore, the TOE design information will be considered as part of the implementation representation.

b) Examination of a particular subset of information in each of the design representations provided for the evaluation.

c) Coverage of particular SFRs through each of the design representations provided for the evaluation.

d) Examination of each of the design representations provided for the evaluation, considering different SFRs within each design representations.

e) Examination of aspects of the evidence provided for the evaluation relating to current potential vulnerability information the evaluator has received (e.g. from a scheme).

1506 This approach to identification of potential vulnerabilities is to take an ordered and planned approach; applying a system to the examination. The evaluator is to describe the method to be used in terms of what evidence will be considered, the information within the evidence that is to be examined, the manner in which this information is to be considered and the hypothesis that is to be created.

1507 The following provide some examples that a hypothesis may take:

a) consideration of malformed input for interfaces available to an attacker at the external interfaces;

b) examination of a key security mechanism cited in the security architecture description, such as process separation, hypothesising internal buffer overflows that may lead to degradation of separation;

c) search to identify any objects created in the TOE implementation representation that are then not fully controlled by the TSF, and could be used by an attacker to undermine SFRs.

1508 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE and specify an approach to the search that “all interface specifications provided in the functional specification and TOE design will be searched to hypothesise potential vulnerabilities” and go on to explain the methods used in the hypothesis.

1509 The identification process is iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1510 The evaluator will report what actions were taken to identify potential vulnerabilities in the evidence. However, in this type of search, the evaluator may not be able to describe the steps in identifying potential vulnerabilities before the outset of the examination, as the approach may evolve as a result of findings during the search.

1511 The evaluator will report the evidence examine in completing the search for potential vulnerabilities. This selection of evidence may be derived from those areas of concern identified by the evaluator, linked to the evidence the attacker is assumed to be able to obtain, or according to another rationale provided by the evaluator.

1512 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent vulnerability analysis should consider generic potential vulnerabilities under each of the following headings:

a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be supplied by the evaluation authority;

b) bypassing;

c) tampering;

d) direct attacks;

e) monitoring;

f) misuse.

1513 Items b) - f) are explained in greater detail in Annex B.

1514 The security architecture description should be considered in light of each of the above generic potential vulnerabilities. Each potential vulnerability should be considered to search for possible ways in which to defeat the TSF protection and undermine the TSF.

AVA_VAN.3-5 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.

1515 It may be identified that no further consideration of the potential vulnerability is required if for example the evaluator identifies that measures in the operational environment, either IT or non-IT, prevent exploitation of the potential vulnerability in that operational environment. For instance, restricting physical access to the TOE to authorised users only may effectively render a potential vulnerability to tampering unexploitable.

1516 The evaluator records any reasons for exclusion of potential vulnerabilities from further consideration if the evaluator determines that the potential vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1517 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15.2.3.7 Action AVA_VAN.3.4E

AVA_VAN.3-6 The evaluator ***shall devise*** penetration tests, based on the independent search for potential vulnerabilities.

1518 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the TOE, in its operational environment, to the potential vulnerabilities identified during the search of the sources of publicly available information and the analysis of the TOE guidance and design evidence. The evaluator should have access to current information (e.g. from the evaluation authority) regarding known potential vulnerabilities that may not have been considered by the evaluator.

1519 The evaluator is reminded that, as for considering the security architecture description in the search for vulnerabilities (as detailed in AVA_VAN.3-3), testing should be performed to confirm the architectural properties. If requirements from ATE_DPT are included in the SARs, the developer testing evidence will include testing performed to confirm the correct implementation of any specific mechanisms detailed in the security architecture description. However, the developer testing will not necessarily include testing of all aspects of the architectural properties that protect the TSF, as much of this testing will be negative testing in nature, attempting to disprove the properties. In developing the strategy for penetration testing, the evaluator will ensure that all aspects of the security architecture description are tested, either in functional testing (as considered in 14) or evaluator penetration testing.

1520 It will probably be practical to carry out penetration test using a series of test cases, where each test case will test for a specific potential vulnerability.

1521 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required an Enhanced-Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Enhanced-Basic attack potential, this is reported in the ETR as a residual vulnerability.

1522 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be found in Annex B.4.

1523 Potential vulnerabilities hypothesised as exploitable only by attackers possessing Moderate or High attack potential do not result in a failure of this evaluator action. Where analysis supports the hypothesis, these need not be considered further as an input to penetration testing. However, such vulnerabilities are reported in the ETR as residual vulnerabilities.

1524 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Basic or Enhanced-Basic attack potential and resulting in a violation of the security objectives should be the highest priority potential vulnerabilities comprising the list used to direct penetration testing against the TOE.

AVA_VAN.3-7 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include:

a) identification of the potential vulnerability the TOE is being tested for;

b) instructions to connect and setup all required test equipment as required to conduct the penetration test;

c) instructions to establish all penetration test prerequisite initial conditions;

d) instructions to stimulate the TSF;

e) instructions for observing the behaviour of the TSF;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE.

1525 The evaluator prepares for penetration testing based on the list of potential vulnerabilities identified during the search of the public domain and the analysis of the evaluation evidence.

1526 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond those for which an Enhanced-Basic attack potential is required to effect an attack. However, as a result of evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only by an attacker with greater than Enhanced-Basic attack potential. Such vulnerabilities are to be reported in the ETR as residual vulnerabilities.

1527 With an understanding of the potential vulnerability, the evaluator determines the most feasible way to test for the TOE's susceptibility. Specifically the evaluator considers:

a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses (It is possible that the evaluator will need to use an interface to the TOE other than the TSFI to demonstrate properties of the TSF such as those described in the security architecture description (as required by ADV_ARC). It should the noted, that although these TOE interfaces provide a means of testing the TSF properties, they are not the subject of the test.);

b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

c) special test equipment that will be required to either stimulate a TSFI or make observations of a TSFI (although it is unlikely that specialist equipment would be required to exploit a potential vulnerability assuming an Enhanced-Basic attack potential);

d) whether theoretical analysis should replace physical testing, particularly relevant where the results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are likely to succeed after a given number of attempts.

1528 The evaluator will probably find it practical to carry out penetration testing using a series of test cases, where each test case will test for a specific potential vulnerability.

1529 The intent of specifying this level of detail in the test documentation is to allow another evaluator to repeat the tests and obtain an equivalent result.

AVA_VAN.3-8 The evaluator ***shall conduct*** penetration testing.

1530 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.3-6 as a basis for executing penetration tests on the TOE, but this does not preclude the evaluator from performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests as a result of information learnt during penetration testing that, if performed by the evaluator, are to be recorded in the penetration test documentation. Such tests may be required to follow up unexpected results or observations, or to investigate potential vulnerabilities suggested to the evaluator during the pre-planned testing.

1531 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the evaluator should determine whether or not the evaluator's own analysis was incorrect, or if evaluation deliverables are incorrect or incomplete.

1532 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required an Enhanced-Basic attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Enhanced-Basic attack potential, this is reported in the ETR as a residual vulnerability.

AVA_VAN.3-9 The evaluator ***shall record*** the actual results of the penetration tests.

1533 While some specific details of the actual test results may be different from those expected (e.g. time and date fields in an audit record) the overall result

should be identical. Any unexpected test results should be investigated. The impact on the evaluation should be stated and justified.

AVA_VAN.3-10 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.

1534 The penetration testing information reported in the ETR allows the evaluator to convey the overall penetration testing approach and effort expended on this sub-activity. The intent of providing this information is to give a meaningful overview of the evaluator's penetration testing effort. It is not intended that the information regarding penetration testing in the ETR be an exact reproduction of specific test steps or results of individual penetration tests. The intention is to provide enough detail to allow other evaluators and evaluation authorities to gain some insight about the penetration testing approach chosen, amount of penetration testing performed, TOE test configurations, and the overall results of the penetration testing activity.

1535 Information that would typically be found in the ETR section regarding evaluator penetration testing efforts is:

a) TOE test configurations. The particular configurations of the TOE that were penetration tested;

b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of the penetration testing;

c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.

1536 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the penetration testing the evaluator performed during the evaluation.

AVA_VAN.3-11 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing an Enhanced-Basic attack potential.

1537 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by an attacker possessing less than Moderate attack potential, then this evaluator action fails.

1538 The guidance in B.4 should be used to determine the attack potential required to exploit a particular vulnerability and whether it can therefore be exploited in the intended environment. It may not be necessary for the attack potential to be calculated in every instance, only if there is some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an attack potential less than Moderate.

AVA_VAN.3-12 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each:

a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);

b) the SFR(s) not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual).

e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.

15.2.4 Evaluation of sub-activity (AVA_VAN.4)

15.2.4.1 Objectives

1539 The objective of this sub-activity is to determine whether the TOE, in its operational environment, has vulnerabilities exploitable by attackers possessing Moderate attack potential.

15.2.4.2 Input

1540 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE design;

d) the security architecture description;

e) the implementation representation;

f) the guidance documentation;

g) the TOE suitable for testing;

h) information publicly available to support the identification of possible potential vulnerabilities.

1541 The remaining implicit evaluation evidence for this sub-activity depends on the components that have been included in the assurance package. The evidence provided for each component is to be used as input in this sub-activity.

1542 Other input for this sub-activity is:

a) current information regarding public domain potential vulnerabilities and attacks (e.g. from an evaluation authority).

15.2.4.3 Application notes

1543 The methodical analysis approach takes the form of a structured examination of the evidence. This method requires the evaluator to specify the structure and form the analysis will take (i.e. the manner in which the analysis is performed is predetermined, unlike the focused analysis). The method is specified in terms of the information that will be considered and how/why it will be considered. Further guidance on methodical vulnerability analysis can be found in Annex B.2.2.2.3.

15.2.4.4 Action AVA_VAN.4.1E

AVA_VAN.4.1C ***The TOE shall be suitable for testing.***

AVA_VAN.4-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

1544 The TOE provided by the developer and identified in the test plan should have the same unique reference as established by the CM capabilities (ALC_CMC) sub-activities and identified in the ST introduction.

1545 It is possible for the ST to specify more than one configuration for evaluation. The TOE may comprise a number of distinct hardware and software entities that need to be tested in accordance with the ST. The evaluator verifies that all test configurations are consistent with the ST.

1546 The evaluator should consider the security objectives for the operational environment described in the ST that may apply to the test environment and ensure they are met in the testing environment. There may be some objectives for the operational environment that do not apply to the test environment. For example, an objective about user clearances may not apply; however, an objective about a single point of connection to a network would apply.

1547 If any test resources are used (e.g. meters, analysers) it will be the evaluator's responsibility to ensure that these resources are calibrated correctly.

AVA_VAN.4-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1548 It is possible for the evaluator to determine the state of the TOE in a number of ways. For example, previous successful completion of the Evaluation of sub-activity (AGD_PRE.1) sub-activity will satisfy this work unit if the evaluator still has confidence that the TOE being used for testing was installed properly and is in a known state. If this is not the case, then the evaluator should follow the developer's procedures to install and start up the TOE, using the supplied guidance only.

1549 If the evaluator has to perform the installation procedures because the TOE is in an unknown state, this work unit when successfully completed could satisfy work unit AGD_PRE.1-5.

15.2.4.5 Action AVA_VAN.4.2E

AVA_VAN.4-3 The evaluator ***shall examine*** sources of information publicly available to identify potential vulnerabilities in the TOE.

1550 The evaluator examines the sources of information publicly available to support the identification of possible potential vulnerabilities in the TOE. There are many sources of publicly available information which the evaluator should consider using items such as those available on the world wide web, including:

a) specialist publications (magazines, books);

b) research papers;

c) conference proceedings.

1551 The evaluator should not constrain their consideration of publicly available information to the above, but should consider any other relevant information available.

1552 While examining the evidence provided the evaluator will use the information in the public domain to further search for potential vulnerabilities. Where the evaluators have identified areas of concern, the evaluator should consider information publicly available that relate to those areas of concern.

1553 The availability of information that may be readily available to an attacker that helps to identify and facilitate attacks may substantially enhance the attack potential of a given attacker. The accessibility of vulnerability information and sophisticated attack tools on the Internet makes it more likely that this information will be used in attempts to identify potential vulnerabilities in the TOE and exploit them. Modern search tools make such information easily available to the evaluator, and the determination of resistance to published potential vulnerabilities and well known generic attacks can be achieved in a cost-effective manner.

1554 The search of the information publicly available should be focused on those sources that refer to the technologies used in the development of the product from which the TOE is derived. The extensiveness of this search should consider the following factors: TOE type, evaluator experience in this TOE type, expected attack potential and the level of ADV evidence available.

1555 The identification process is iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1556 The evaluator will describe the approach to be taken to identify potential vulnerabilities in the publicly available material, detailing the search to be performed. This may be driven by factors such as areas of concern identified by the evaluator, linked to the evidence the attacker is assumed to be able to

obtain. However, it is recognised that in this type of search the approach may further evolve as a result of findings during the search. Therefore, the evaluator will also report any actions taken in addition to those described in the approach to further investigate issues thought to lead to potential vulnerabilities, and will report the evidence examined in completing the search for potential vulnerabilities.

15.2.4.6 Action AVA_VAN.4.3E

AVA_VAN.4-4 The evaluator ***shall conduct*** a methodical analysis of ST, guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify possible potential vulnerabilities in the TOE.

1557 Guidance on methodical vulnerability analysis is provided in Annex B.2.2.2.3.

1558 This approach to identification of potential vulnerabilities is to take an ordered and planned approach. A system is to be applied in the examination. The evaluator is to describe the method to be used in terms of the manner in which this information is to be considered and the hypothesis that is to be created.

1559 A flaw hypothesis methodology should be used whereby the ST, development (functional specification, TOE design and implementation representation) and guidance evidence are analysed and then vulnerabilities in the TOE are hypothesised, or speculated.

1560 The evaluator should use the knowledge of the TOE design and operation gained from the TOE deliverables to conduct a flaw hypothesis to identify potential flaws in the development of the TOE and potential errors in the specified method of operation of the TOE.

1561 The security architecture description provides the developer vulnerability analysis, as it documents how the TSF protects itself from interference from untrusted subjects and prevents the bypass of security enforcement functionality. Therefore, the evaluator should build upon the understanding of the TSF protection gained from the analysis of this evidence and then develop this in the knowledge gained from other development (11 evidence.

1562 The approach taken to the methodical search for vulnerabilities is to consider any areas of concern identified in the results of the evaluator's assessment of the development and guidance evidence. However, the evaluator should also consider each aspect of the security architecture analysis to search for any ways in which the protection of the TSF can be undermined. It may be helpful to structure the methodical analysis on the basis of the material presented in the security architecture description, introducing concerns from other 11 evidence as appropriate. The analysis can then be further developed to ensure all other material from the 11 evidence is considered.

1563 The following provide some examples of hypotheses that may be created when examining the evidence:

a) consideration of malformed input for interfaces available to an attacker at the external interfaces;

b) examination of a key security mechanism cited in the security architecture description, such as process separation, hypothesising internal buffer overflows that may lead to degradation of separation;

c) search to identify any objects created in the TOE implementation representation that are then not fully controlled by the TSF, and could be used by an attacker to undermine SFRs.

1564 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE and specify an approach to the search that 'all interface specifications in the evidence provided will be searched to hypothesise potential vulnerabilities' and go on to explain the methods used in the hypothesis.

1565 In addition, areas of concern the evaluator has identified during examination of the evidence during the conduct of evaluation activities. Areas of concern may also be identified during the conduct of other work units associated with this component, in particular AVA_VAN.4-7, AVA_VAN.4-5 and AVA_VAN.4-6) where the development and conduct of penetration tests may identify further areas of concerns for investigation, or potential vulnerabilities.

1566 However, examination of only a subset of the development and guidance evidence or their contents is not permitted in this level of rigour. The approach description should provide a demonstration that the methodical approach used is complete, providing confidence that the approach used to search the deliverables has considered all of the information provided in those deliverables.

1567 This approach to identification of potential vulnerabilities is to take an ordered and planned approach; applying a system to the examination. The evaluator is to describe the method to be used in terms of how the evidence will be considered; the manner in which this information is to be considered and the hypothesis that is to be created. This approach should be agreed with the evaluation authority, and the evaluation authority should provide detail of any additional approaches the evaluator should take to the vulnerability analysis and identify any additional information that should be considered by the evaluator.

1568 Although a system to identifying potential vulnerabilities is predefined, the identification process may still be iterative, where the identification of one potential vulnerability may lead to identifying another area of concern that requires further investigation.

1569 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent vulnerability analysis should consider generic potential vulnerabilities under each of the following headings:

a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be supplied by the evaluation authority;

b) bypassing;

c) tampering;

d) direct attacks;

e) monitoring;

f) misuse.

1570 Items b) - f) are explained in greater detail in Annex B.

1571 The security architecture description should be considered in light of each of the above generic potential vulnerabilities. Each potential vulnerability should be considered to search for possible ways in which to defeat the TSF protection and undermine the TSF.

AVA_VAN.4-5 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.

1572 It may be identified that no further consideration of the potential vulnerability is required if for example the evaluator identifies that measures in the operational environment, either IT or non-IT, prevent exploitation of the potential vulnerability in that operational environment. For instance, restricting physical access to the TOE to authorised users only may effectively render a potential vulnerability to tampering unexploitable.

1573 The evaluator records any reasons for exclusion of potential vulnerabilities from further consideration if the evaluator determines that the potential vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1574 A list of potential vulnerabilities applicable to the TOE in its operational environment, which can be used as an input into penetration testing activities, shall be reported in the ETR by the evaluators.

15.2.4.7 Action AVA_VAN.4.4E

AVA_VAN.4-6 The evaluator ***shall devise*** penetration tests, based on the independent search for potential vulnerabilities.

1575 The evaluator prepares for penetration testing as necessary to determine the susceptibility of the TOE, in its operational environment, to the potential vulnerabilities identified during the search of the sources of publicly

available information and the analysis of the TOE guidance and design evidence. The evaluator should have access to current information (e.g. from the evaluation authority) regarding known potential vulnerabilities that may not have been considered by the evaluator.

1576 The evaluator is reminded that, as for considering the security architecture description in the search for vulnerabilities (as detailed in AVA_VAN.4-3), testing should be performed to confirm the architectural properties. If requirements from ATE_DPT are included in the SARs, the developer testing evidence will include testing performed to confirm the correct implementation of any specific mechanisms detailed in the security architecture description. However, the developer testing will not necessarily include testing of all aspects of the architectural properties that protect the TSF, as much of this testing will be negative testing in nature, attempting to disprove the properties. In developing the strategy for penetration testing, the evaluator will ensure that all aspects of the security architecture description are tested, either in functional testing (as considered in 14) or evaluator penetration testing.

1577 The evaluator will probably find it practical to carry out penetration test using a series of test cases, where each test case will test for a specific potential vulnerability.

1578 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Moderate attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Moderate attack potential, this is reported in the ETR as a residual vulnerability.

1579 Guidance on determining the necessary attack potential to exploit a potential vulnerability can be found in Annex B.4.

1580 Potential vulnerabilities hypothesised as exploitable by an attacker possessing a Moderate (or less) attack potential and resulting in a violation of the security objectives should be the highest priority potential vulnerabilities comprising the list used to direct penetration testing against the TOE.

AVA_VAN.4-7 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include:

a) identification of the potential vulnerability the TOE is being tested for;

b) instructions to connect and setup all required test equipment as required to conduct the penetration test;

c) instructions to establish all penetration test prerequisite initial conditions;

d) instructions to stimulate the TSF;

e) instructions for observing the behaviour of the TSF;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE.

1581 The evaluator prepares for penetration testing based on the list of potential vulnerabilities identified during the search of the public domain and the analysis of the evaluation evidence.

1582 The evaluator is not expected to determine the exploitability for potential vulnerabilities beyond those for which a Moderate attack potential is required to effect an attack. However, as a result of evaluation expertise, the evaluator may discover a potential vulnerability that is exploitable only by an attacker with greater than Moderate attack potential. Such vulnerabilities are to be reported in the ETR as residual vulnerabilities.

1583 With an understanding of the potential vulnerability, the evaluator determines the most feasible way to test for the TOE's susceptibility. Specifically the evaluator considers:

a) the TSFI or other TOE interface that will be used to stimulate the TSF and observe responses (It is possible that the evaluator will need to use an interface to the TOE other than the TSFI to demonstrate properties of the TSF such as those described in the security architecture description (as required by ADV_ARC). It should the noted, that although these TOE interfaces provide a means of testing the TSF properties, they are not the subject of the test.);

b) initial conditions that will need to exist for the test (i.e. any particular objects or subjects that will need to exist and security attributes they will need to have);

c) special test equipment that will be required to either stimulate a TSFI or make observations of a TSFI;

d) whether theoretical analysis should replace physical testing, particularly relevant where the results of an initial test can be extrapolated to demonstrate that repeated attempts of an attack are likely to succeed after a given number of attempts.

1584 The evaluator will probably find it practical to carry out penetration testing using a series of test cases, where each test case will test for a specific potential vulnerability.

1585 The intent of specifying this level of detail in the test documentation is to allow another evaluator to repeat the tests and obtain an equivalent result.

AVA_VAN.4-8 The evaluator ***shall conduct*** penetration testing.

1586 The evaluator uses the penetration test documentation resulting from work unit AVA_VAN.4-6 as a basis for executing penetration tests on the TOE, but this does not preclude the evaluator from performing additional ad hoc penetration tests. If required, the evaluator may devise ad hoc tests as a result of information learnt during penetration testing that, if performed by the evaluator, are to be recorded in the penetration test documentation. Such tests may be required to follow up unexpected results or observations, or to investigate potential vulnerabilities suggested to the evaluator during the pre-planned testing.

1587 Should penetration testing show that a hypothesised potential vulnerability does not exist, then the evaluator should determine whether or not the evaluator's own analysis was incorrect, or if evaluation deliverables are incorrect or incomplete.

1588 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those which required a Moderate attack potential. In some cases, however, it will be necessary to carry out a test before the exploitability can be determined. Where, as a result of evaluation expertise, the evaluator discovers an exploitable vulnerability that is beyond Moderate attack potential, this is reported in the ETR as a residual vulnerability.

AVA_VAN.4-9 The evaluator ***shall record*** the actual results of the penetration tests.

1589 While some specific details of the actual test results may be different from those expected (e.g. time and date fields in an audit record) the overall result should be identical. Any unexpected test results should be investigated. The impact on the evaluation should be stated and justified.

AVA_VAN.4-10 The evaluator ***shall report*** in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.

1590 The penetration testing information reported in the ETR allows the evaluator to convey the overall penetration testing approach and effort expended on this sub-activity. The intent of providing this information is to give a meaningful overview of the evaluator's penetration testing effort. It is not intended that the information regarding penetration testing in the ETR be an exact reproduction of specific test steps or results of individual penetration tests. The intention is to provide enough detail to allow other evaluators and evaluation authorities to gain some insight about the penetration testing approach chosen, amount of penetration testing performed, TOE test configurations, and the overall results of the penetration testing activity.

1591 Information that would typically be found in the ETR section regarding evaluator penetration testing efforts is:

a) TOE test configurations. The particular configurations of the TOE that were penetration tested;

b) TSFI penetration tested. A brief listing of the TSFI and other TOE interfaces that were the focus of the penetration testing;

c) Verdict for the sub-activity. The overall judgement on the results of penetration testing.

1592 This list is by no means exhaustive and is only intended to provide some context as to the type of information that should be present in the ETR concerning the penetration testing the evaluator performed during the evaluation.

AVA_VAN.4-11 The evaluator ***shall examine*** the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Moderate attack potential.

1593 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by an attacker possessing less than a High attack potential, then this evaluator action fails.

1594 The guidance in B.4 should be used to determine the attack potential required to exploit a particular vulnerability and whether it can therefore be exploited in the intended environment. It may not be necessary for the attack potential to be calculated in every instance, only if there is some doubt as to whether or not the vulnerability can be exploited by an attacker possessing an attack potential less than High.

AVA_VAN.4-12 The evaluator ***shall report*** in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each:

a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);

b) the SFR(s) not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual).

e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.

15.2.5 Evaluation of sub-activity (AVA_VAN.5)

1595 There is no general guidance; the scheme should be consulted for guidance on this sub-activity.

16 Class ACO: Composition

16.1 Introduction

1596 The goal of this activity is to determine whether the components can be integrated in a secure manner, as defined in the ST for the composed TOE. This is achieved through examination and testing of the interfaces between the components, supported by examination of the design of the components and the conduct of vulnerability analysis.

16.2 Application notes

1597 The Reliance of dependent component (ACO_REL) family identifies where the dependent component is reliant upon IT in its operational environment (satisfied by a base component in the composed TOE evaluation) in order to provide its own security services. This reliance is identified in terms of the interfaces expected by the dependent component to be provided by the base component. Development evidence (ACO_DEV) then determines which interfaces of the base component were considered (as TSFI) during the component evaluation of the base component.

1598 It should be noted that Reliance of dependent component (ACO_REL) does not cover other evidence that may be needed to address the technical integration problem of composing components (e.g. descriptions of non-TSF interfaces of the operating system, rules for integration, etc.). This is outside the security assessment of the composition and is a functional composition issue.

1599 As part of Composed TOE testing (ACO_CTT) the evaluator will perform testing of the composed TOE SFRs at the composed TOE interfaces and of the interfaces of the base component relied upon by the dependent component to confirm they operate as specified. The subset selected will consider the possible effects of changes to the configuration/use of the base component as used in the composed TOE. These changes are identified from the configuration of the base component determined during the base component evaluation. The developer will provide test evidence for each of the base component interfaces (the requirements for coverage are consistent with those applied to the evaluation of the base component).

1600 Composition rationale (ACO_COR) requires the evaluator to determine whether the appropriate assurance measures have been applied to the base component, and whether the base component is being used in its evaluated configuration. This includes determination of whether all security functionality required by the dependent component was within the TSF of the base component. The Composition rationale (ACO_COR) requirement may be met through the production of evidence that each of these is demonstrated to be upheld. This evidence may be in the form of the security target and a public report of the component evaluation (e.g. certification report).

1601 If, on the other hand, one of the above have not been upheld, then it may be possible that an argument can be made as to why the assurance gained during an original evaluation is unaffected. If this is not possible then additional evaluation evidence for those aspects of the base component not covered may have to be provided. This material is then assessed in Development evidence (ACO_DEV).

1602 For example, it may be the case as described in the Interactions between entities (see Annex B.3, Interactions between composed IT entities in CC Part 3) that the dependent component requires the base component to provide more security functionality in the composed TOE than included in the base component evaluation. This would be determined during the application of the Reliance of dependent component (ACO_REL) and Development evidence (ACO_DEV) families. In this case the composition rationale evidence provided for Composition rationale (ACO_COR) would demonstrate that the assurance gained from the base component evaluation is unaffected. This may be achieved by means including:

a) Performing a re-evaluation of the base component focusing on the evidence relating to the extended part of the TSF;

b) Demonstrating that the extended part of the TSF cannot affect other portions of the TSF, and providing evidence that the extended part of the TSF provides the necessary security functionality.

16.3 Composition rationale (ACO_COR)

16.3.1 Evaluation of sub-activity (ACO_COR.1)

16.3.1.1 Input

1603 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the composition rationale;

c) the reliance information;

d) the development information;

e) unique identifier.

16.3.1.2 Action ACO_COR.1.1E

ACO_COR.1.1C ***The composition rationale shall demonstrate that a level of assurance at least as high as that of the dependent component has been obtained for the support functionality of the base component, when the base component is configured as required to support the TSF of the dependent component.***

ACO_COR.1-1 The evaluator ***shall examine*** the correspondence analysis with the development information and the reliance information to identify the interfaces that are relied upon by the dependent component which are not detailed in the development information.

1604 The evaluator's goal in this work unit is two fold:

a) to determine which interfaces relied upon by the dependent component have had the appropriate assurance measures applied.

b) to determine that the assurance package applied to the base component during the base component evaluation contained either the same assurance requirements as those in the package applied to the dependent component during its' evaluation, or hierarchically higher assurance requirements.

1605 The evaluator may use the correspondence tracing in the development information developed during the Development evidence (ACO_DEV) activities (e.g. ACO_DEV.1-2, ACO_DEV.2-4, ACO_DEV.3-6) to help identify the interfaces identified in the reliance information that are not considered in the development information.

1606 The evaluator will record the SFR-enforcing interfaces described in the reliance information that are not included in the development information. These will provide input to ACO_COR.1-3 work unit, helping to identify the portions of the base component in which further assurance is required.

1607 If the both the base and dependent components were evaluated against the same assurance package, then the determination of whether the level of assurance in the portions within the base component evaluation is at least as high as that of the dependent component is trivial. If however, the assurance packages applied to the components during the component evaluations differ, the evaluator needs to determine that the assurance requirements applied to the base component are all hierarchically higher to the assurance requirements applied to the dependent component.

ACO_COR.1-2 The evaluator ***shall examine*** the composition rationale to determine, for those included base component interfaces on which the dependent TSF relies, whether the interface was considered during the evaluation of the base component.

1608 The ST, component public evaluation report (e.g. certification report) and guidance documents for the base component all provide information on the scope and boundary of the base component. The ST provides details of the logical scope and boundary of the composed TOE, allowing the evaluator to determine whether an interface relates to a portion of the product that was within the scope of the evaluation. The guidance documentation provides details of use of all interfaces for the composed TOE. Although the guidance documentation may include details of interfaces in the product that are not within the scope of the evaluation, any such interfaces should be identifiable, either from the scoping information in the ST or through a portion of the guidance that deals with the evaluated configuration. The public evaluation report should provide any additional constraints on the use of the composed TOE that are necessary.

1609 Therefore, the combination of these inputs allows the evaluator to determine whether an interface described in the composition rationale has the necessary assurance associated with it, or whether further assurance is required. The evaluator will record those interfaces of the base component for which additional assurance is required, for consideration during ACO_COR.1-3.

ACO_COR.1-3 The evaluator ***shall examine*** the composition rationale to determine that the necessary assurance measures have been applied to the base component.

1610 The evaluation verdicts, and resultant assurance, for the base component can be reused provided the same portions of the base component are used in the composed TOE and they are used in a consistent manner.

1611 In order to determine whether the necessary assurance measures have already been applied to the component, and the portions of the component for which assurance measures still need to be applied, the evaluator should use the output of the ACO_DEV.*.2E action and the work units ACO_COR.1-1 and ACO_COR.1-2:

a) For those interfaces identified in the reliance information (Reliance of dependent component (ACO_REL)), but not discussed in development information (Development evidence (ACO_DEV)), additional information is required. (Identified in ACO_COR.1-1.)

b) For those interfaces used inconsistently in the composed TOE from the base component (difference between the information provided in Development evidence (ACO_DEV) and Reliance of dependent component (ACO_REL) the impact of the differences in use need to be considered. (Identified in ACO_DEV.*.2E.)

c) For those interfaces identified in composition rationale for which no assurance has previously been gained, additional information is required. (Identified in ACO_COR.1-2.)

d) For those interfaces consistently described in the reliance information, composition rationale and the development information, no further action is required as the results from the base component evaluation can be re-used.

1612 The interfaces of the base component reported to be required by the reliance information but not included in the development information indicate the portions of the base component where further assurance is required. The interfaces identify the entry points into the base component.

1613 For those interfaces included in both the development information and reliance information, the evaluator is to determine whether the interfaces are being used in the composed TOE in a manner that is consistent with the base component evaluation. The method of use of the interface will be considered during the Development evidence (ACO_DEV) activities to determine that the use of the interface is consistent in both the base component and the composed TOE. The remaining consideration is the determination of whether the configurations of the base component and the composed TOE are consistent. To determine this, the evaluator will consider the guidance documentation of each to ensure they are consistent (see further guidance below regarding consistent guidance documentation). Any deviation in the documentation will be further analysed by the evaluation to determine the possible effects.

1614 For those interfaces that are consistently described in the reliance information and development information, and for which the guidance is consistent for the base component and the composed TOE, the required level of assurance has been provided.

1615 The following subsections provide guidance on how to determine consistency between assurance gained in the base component, the evidence provided for the composed TOE, and the analysis performed by the evaluator in the instances where inconsistencies are identified.

16.3.1.2.1 Development

1616 The reliance information identifies the interfaces in the dependent component that are to be matched by the base component. If an interface identified in the reliance information is not identified in the development information, then the composition rationale is to provide a justification of how the base component provides the required interfaces.

1617 If an interface identified in the reliance information is identified in the development information, but there are inconsistencies between the descriptions, further analysis is required. The evaluator identifies the differences in use of the base component as considered in the base component evaluation and the composed TOE evaluation. The evaluator will devise testing to be performed (during the conduct of Composed TOE testing (ACO_CTT)) to test the interface.

1618 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.

1619 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)).

16.3.1.2.2 Guidance

1620 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.

1621 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.

1622 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.

1623 *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.

1624 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 TOE design (ADV_TDS) requirements) it is possible that, unless detailed design abstractions for the base component are delivered as part of the development information for the composed TOE, the possible impacts of the modification to the guidance cannot be fully determined as the internals are unknown. In this case the evaluator will report the residual risk of the analysis.

1625 These residual risks are to be included in any public evaluation report for the composed TOE.

1626 The evaluator will note these variances in the guidance for input into evaluator independent testing activities (Composed TOE testing (ACO_CTT)).

1627 The guidance for the composed TOE may add to the guidance for the components, particularly in terms of installation and the ordering of installation steps for the base component in relation to the installation steps for the dependent component. The ordering of the steps for the installation of the individual components should not change, however they may need to be interleaved. The evaluator will examine this guidance to ensure that it still meets the requirement of the AGD_PRE activity performed during the evaluations of the components.

1628 It may be the case that the reliance information identifies that interfaces of the base component, in addition to those identified as TSFIs of the base component, are relied upon by the dependent 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.

1629 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.

1630 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.*

1631 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.

1632 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.1-5 during the dependent component evaluation, the evaluator will apply the user procedures provided for the composed TOE to prepare the composed TOE, in accordance with the guidance specified in AGD_PRE.1-5. This will allow the evaluator to determine that the preparative guidance provided for the composed TOE is sufficient to prepare the composed TOE and its operational environment securely.

16.3.1.2.3 Life-cycle

Delivery

1633 If there is a different delivery mechanism used for the delivery of the composed TOE (i.e. the components are not delivered to the consumer in accordance with the secure delivery procedures defined and assessed during the evaluation of the components), the delivery procedures for the composed TOE will require evaluation against the Delivery (ALC_DEL) requirements applied during the components evaluations.

1634 The composed TOE may be delivered as an integrated product or may require the components to be delivered separately.

1635 If the components are delivered separately, the results of the delivery of the base component and dependent component are reused. The delivery of the base component is checked during the evaluator trial installation of the dependent component, using the specified guidance and checking the aspects of delivery that are the responsibility of the user, as described in the guidance documentation for the base component.

1636 If the composed TOE is delivered as a new entity, then the method of delivery of that entity must be considered in the composed TOE evaluation activities.

1637 The assessment of the delivery procedures for composed TOE items is to be performed in accordance with the methodology for Delivery (ALC_DEL) as for any other [component] TOE, ensuring any additional items (e.g. additional guidance documents for the composed TOE) are considered in the delivery procedures.

CM Capabilities

1638 The unique identification of the composed TOE is considered during the application of Evaluation of sub-activity (ALC_CMC.1) and the items from which that composed TOE is comprised are considered during the application of Evaluation of sub-activity (ALC_CMS.2).

1639 Although additional guidance may be produced for the composed TOE, the unique identification of this guidance (considered as part of the unique identification of the composed TOE during Evaluation of sub-activity (ALC_CMC.1)) is considered sufficient control of the guidance.

1640 The verdicts of the remaining (not considered above) Class ALC: Life-cycle support activities can be reused from the base component evaluation, as no further development is performed during integration of the composed TOE.

1641 There are no additional considerations for development security as the integration is assumed to take place at either the consumer's site or, in the instance that the composed TOE is delivered as an integrated product, at the site of the dependent component developer. Control at the consumer's site is outside the consideration of the CC. No additional requirements or guidance are necessary if integration is at the same site as that for the dependent component, as all components are considered to be configuration items for the composed TOE, and should therefore be considered under the dependent component developer's security procedures anyway.

1642 Tools and techniques adopted during integration will be considered in the evidence provided by the dependent component developer. Any tools/techniques relevant to the base component will have been considered during the evaluation of the base component. For example, if the base component is delivered as source code and requires compilation by the consumer (e.g. dependent component developer who is performing integration) the compiler would have been specified and assessed, along with the appropriate arguments, during evaluation of the base component.

1643 There is no life-cycle definition applicable to the composed TOE, as no further development of items is taking place.

1644 The results of flaw remediation for a component are not applicable to the composed TOE. If flaw remediation is included in the assurance package for the composed TOE, then the Flaw remediation (ALC_FLR) requirements are to be applied during the composed TOE evaluation (as for any augmentation).

16.3.1.2.4 Tests

1645 The composed TOE will have been tested during the conduct of the Class ATE: Tests activities for evaluation of the dependent component, as the configurations used for testing of the dependent component should have included the base component to satisfy the requirements for IT in the operational environment. If the base component was not used in the testing of the dependent component for the dependent component evaluation, or the configuration of either component varied from their evaluated configurations, then the developer testing performed for evaluation of the dependent component to satisfy the Class ATE: Tests requirements is to be repeated on the composed TOE.

16.4 Development evidence (ACO_DEV)

16.4.1 Evaluation of sub-activity (ACO_DEV.1)

16.4.1.1 Objectives

1646 The objective of this sub-activity is to determine that the appropriate security functionality is provided by the base component to support the dependent component. This is achieved through examination of the interfaces of the base component to determine that they are consistent with the interfaces specified in the reliance information; those required by the dependent component.

1647 The description of the interfaces into the base component is to be provided at a level of detail consistent with Evaluation of sub-activity (ADV_FSP.2) although not all of the aspects necessary for satisfaction of Evaluation of sub-activity (ADV_FSP.2) are required for Evaluation of sub-activity (ACO_DEV.1), as once the interface has been identified and the purpose described the remaining detail of the interface specification can be reused from evaluation of the base component.

16.4.1.2 Input

1648 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the development information;

c) the reliance information.

16.4.1.3 Action ACO_DEV.1.1E

ACO_DEV.1.1C ***The development information shall describe the purpose of each interface of the base component used in the composed TOE.***

ACO_DEV.1-1 The evaluator ***shall examine*** the development information to determine that it describes the purpose of each interface.

1649 The base component provides interfaces to support interaction with the dependent component in the provision of the dependent TSF. The purpose of each interface is to be described at the same level as the description of the interfaces to the dependent component TSF functionality, as would be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This description is to provide the reader with an understanding of how the base component provides the services required by the dependent component TSF.

1650 This work unit may be satisfied by the provision of the functional specification for the base component for those interfaces that are TSFIs of the base component.

ACO_DEV.1.2C ***The development information shall show correspondence between the interfaces, used in the composed TOE, of the base component and the dependent component to support the TSF of the dependent component.***

ACO_DEV.1-2 The evaluator ***shall examine*** the development information to determine the correspondence, between the interfaces of the base component and the interfaces on which the dependent component relies, is accurate.

1651 The correspondence between the interfaces of the base component and the interfaces on which the dependent component relies may take the form of a matrix or table. The interfaces that are relied upon by the dependent component are identified in the reliance information (as examined during Reliance of dependent component (ACO_REL) activity).

1652 There is, during this activity, no requirement to determine completeness of the coverage of interfaces that are relied upon by the dependent component, only that the correspondence is correct and ensuring that interfaces of the base component are mapped to interfaces required by the dependent component wherever possible. The completeness of the coverage is considered in Composition rationale (ACO_COR) activities.

16.4.1.4 Action ACO_DEV.1.2E

ACO_DEV.1-3 The evaluator ***shall examine*** the development information and the reliance information to determine that the interfaces are described consistently.

1653 The evaluator's goal in this work unit is to determine that the interfaces described in the development information for the base component and the reliance information for the dependent component are represented consistently.

16.4.2 Evaluation of sub-activity (ACO_DEV.2)

16.4.2.1 Objectives

1654 The objective of this sub-activity is to determine that the appropriate security functionality is provided by the base component to support the dependent component. This is achieved through examination of the interfaces and associated security behaviour of the base component to determine that they are consistent with the interfaces specified in the reliance information; those required by the dependent component.

16.4.2.2 Input

1655 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the development information;

c) reliance information.

16.4.2.3 Action ACO_DEV.2.1E

ACO_DEV.2.1C ***The development information shall describe the purpose and method of use of each interface of the base component used in the composed TOE.***

ACO_DEV.2-1 The evaluator ***shall examine*** the development information to determine that it describes the purpose of each interface.

1656 The base component provides interfaces to support interaction with the dependent component in the provision of the dependent TSF. The purpose of each interface is to be described at the same level as the description of the interfaces to the dependent component TSF functionality, as would be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This description is to provide the reader with an understanding of how the base component provides the services required by the dependent component TSF.

1657 This work unit may be satisfied by the provision of the functional specification for the base component for those interfaces that are TSFIs of the base component.

ACO_DEV.2-2 The evaluator ***shall examine*** the development information to determine that it describes the method of use for each interface.

1658 The method of use for an interface summarises how the interface is manipulated in order to invoke the operations and obtain results associated with the interface. The evaluator should be able to determine from reading this material in the development information how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each interface, as it may be possible to describe in general how APIs are invoked, for instance, and then identify each interface using that general style.

1659 This work unit may be satisfied by the provision of the functional specification for the base component for those interfaces that are TSFIs of the base component.

ACO_DEV.2.2C ***The development information shall provide a high-level description of the behaviour of the base component, which supports the enforcement of the dependent component SFRs.***

ACO_DEV.2-3 The evaluator ***shall examine*** the development information to determine that it describes the behaviour of the base component that supports the enforcement of the dependent component SFRs.

1660 The dependent component invokes interfaces of the base component for the provision of services by the base component. For the interfaces of the base component that are invoked, the development information shall provide a high-level description of the associated security behaviour of the base component. The description of the base component security behaviour will outline how the base component provides the necessary service when the call

to the interface is made. This description is to be at a level similar to that provided for ADV_TDS.1.4C. Therefore, the provision of the TOE design evidence from the base component evaluation would satisfy this work unit, where the interfaces invoked by the dependent component are TSFI of the base component. If the interfaces invoked by the dependent component are not TSFIs of the base component it is the associated security behaviour will not necessarily be described in the base component TOE design evidence.

ACO_DEV.2.3C ***The development information shall show correspondence between the interfaces, used in the composed TOE, of the base component and the dependent component to support the TSF of the dependent component.***

ACO_DEV.2-4 The evaluator ***shall examine*** the development information to determine the correspondence, between the interfaces of the base component and the interfaces on which the dependent component relies, is accurate.

1661 The correspondence between the interfaces of the base component and the interfaces on which the dependent component relies may take the form of a matrix or table. The interfaces that are relied upon by the dependent component are identified in the reliance information (as examined during Reliance of dependent component (ACO_REL)).

1662 There is, during this activity, no requirement to determine completeness of the coverage of interfaces that are relied upon by the dependent component, only that the correspondence is correct and ensuring that interfaces of the base component are mapped to interfaces required by the dependent component wherever possible. The completeness of the coverage is considered in Composition rationale (ACO_COR) activities.

16.4.2.4 Action ACO_DEV.2.2E

ACO_DEV.2-5 The evaluator ***shall examine*** the development information and the reliance information to determine that the interfaces are described consistently.

1663 The evaluator's goal in this work unit is to determine that the interfaces described in the development information for the base component and the reliance information for the dependent component are represented consistently.

16.4.3 Evaluation of sub-activity (ACO_DEV.3)

16.4.3.1 Objectives

1664 The objective of this sub-activity is to determine that the appropriate security functionality is provided by the base component to support the dependent component. This is achieved through examination of the interfaces and associated security behaviour of the base component to determine that they are consistent with the interfaces specified in the reliance information; those required by the dependent component.

1665 In addition to the interface description, the subsystems of the base component that provide the security functionality required by the dependent component will be described to enable the evaluator to determine whether or not that interface formed part of the TSF of the base component.

16.4.3.2 Input

1666 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the development information;

c) reliance information.

16.4.3.3 Action ACO_DEV.3.1E

ACO_DEV.3.1C ***The development information shall describe the purpose and method of use of each interface of the base component used in the composed TOE.***

ACO_DEV.3-1 The evaluator ***shall examine*** the development information to determine that it describes the purpose of each interface.

1667 The base component provides interfaces to support interaction with the dependent component in the provision of the dependent TSF. The purpose of each interface is to be described at the same level as the description of the interfaces to the dependent component TSF functionality, as would be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)). This description is to provide the reader with an understanding of how the base component provides the services required by the dependent component TSF.

1668 This work unit may be satisfied by the provision of the functional specification for the base component for those interfaces that are TSFIs of the base component.

ACO_DEV.3-2 The evaluator ***shall examine*** the development information to determine that it describes the method of use for each interface.

1669 The method of use for an interface summarises how the interface is manipulated in order to invoke the operations and obtain results associated with the interface. The evaluator should be able to determine from reading this material in the development information how to use each interface. This does not necessarily mean that there needs to be a separate method of use for each interface, as it may be possible to describe in general how APIs are invoked, for instance, and then identify each interface using that general style.

1670 This work unit may be satisfied by the provision of the functional specification for the base component for those interfaces that are TSFIs of the base component.

ACO_DEV.3.2C ***The development information shall identify the subsystems of the base component that provide interfaces of the base component used in the composed TOE.***

ACO_DEV.3-3 The evaluator ***shall examine*** the development information to determine that all subsystems of the base component that provide interfaces to the dependent component are identified.

1671 For those interfaces that are considered to form part of the TSFI of the base component, the subsystems associated with the interface will be subsystems considered in the TOE design (ADV_TDS) activity during the base component evaluation. The interfaces on which the dependent component relies that did not form part of the TSFI of the base component will map to subsystems outside of the base component TSF.

ACO_DEV.3.3C ***The development information shall provide a high-level description of the behaviour of the base component subsystems, which support the enforcement of the dependent component SFRs.***

ACO_DEV.3-4 The evaluator ***shall examine*** the development information to determine that it describes the behaviour of the base component subsystems that support the enforcement of the dependent component SFRs.

1672 The dependent component invokes interfaces of the base component for the provision of services by the base component. For the interfaces of the base component that are invoked, the development information shall provide a high-level description of the associated security behaviour of the base component. The description of the base component security behaviour will outline how the base component provides the necessary service when the call to the interface is made. This description is to be at a level similar to that provided for ADV_TDS.1.4C. Therefore, the provision of the TOE design evidence from the base component evaluation would satisfy this work unit, where the interfaces invoked by the dependent component are TSFI of the base component. If the interfaces invoked by the dependent component are not TSFIs of the base component it is the associated security behaviour will not necessarily be described in the base component TOE design evidence.

ACO_DEV.3.4C ***The development information shall provide a mapping from the interfaces to the subsystems of the base component.***

ACO_DEV.3-5 The evaluator ***shall examine*** the development information to determine that the correspondence between the interfaces and subsystems of the base component is accurate.

1673 If the TOE design and functional specification evidence from the base component evaluation is available, this can be used to verify the accuracy of the correspondence between the interfaces and subsystems of the base component as used in the composed TOE. Those interfaces of the base component, which formed part of the base component TSFI will be described in the base component functional specification, and the associated subsystems will be described in the base component TOE design evidence.

The tracing between the two will be provided in the base component TOE design evidence.

1674 If, however, the base component interface did not form part of the TSFI of the base component, the description of the subsystem behaviour provided in the development information will be used to verify the accuracy of the correspondence.

ACO_DEV.3.5C ***The development information shall show correspondence between the interfaces, used in the composed TOE, of the base component and the dependent component to support the TSF of the dependent component.***

ACO_DEV.3-6 The evaluator ***shall examine*** the development information to determine the correspondence, between the interfaces of the base component and the interfaces on which the dependent component relies, is accurate.

1675 The correspondence between the interfaces of the base component and the interfaces on which the dependent component relies may take the form of a matrix or table. The interfaces that are relied upon by the dependent component are identified in the reliance information (as examined during Reliance of dependent component (ACO_REL)).

1676 There is, during this activity, no requirement to determine completeness of the coverage of interfaces that are relied upon by the dependent component, only that the correspondence is correct and ensuring that interfaces of the base component are mapped to interfaces required by the dependent component wherever possible. The completeness of the coverage is considered in Composition rationale (ACO_COR) activities.

16.4.3.4 Action ACO_DEV.3.2E

ACO_DEV.3-7 The evaluator ***shall examine*** the development information and the reliance information to determine that the interfaces are described consistently.

1677 The evaluator's goal in this work unit is to determine that the interfaces described in the development information for the base component and the reliance information for the dependent component are represented consistently.

16.5 Reliance of dependent component (ACO_REL)

16.5.1 Evaluation of sub-activity (ACO_REL.1)

16.5.1.1 Objectives

1678 The objectives of this sub-activity are to determine whether the developer's reliance evidence provides sufficient information to determine that the necessary functionality is available in the base component, and the means by which that functionality is invoked. These are provided in terms of a high-level description.

16.5.1.2 Input

1679 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the dependent component functional specification;

c) the dependent component design;

d) the dependent component architectural design;

e) the reliance information.

16.5.1.3 Application notes

1680 A dependent component whose TSF interacts with the base component requires functionality provided by that base component (e.g., remote authentication, remote audit data storage). In these cases, those invoked services need to be described for those charged with configuring the composed TOE for end users. The rationale for requiring this documentation is to aid integrators of the composed TOE to determine what services in the base component might have adverse effects on the dependent component, and to provide information against which to determine the compatibility of the components when applying the Development evidence (ACO_DEV) family.

16.5.1.4 Action ACO_REL.1.1E

ACO_REL.1.1C ***The reliance information shall describe the functionality of the base component hardware, firmware and/or software that is relied upon by the dependent component TSF.***

ACO_REL.1-1 The evaluator ***shall check*** the reliance information to determine that it describes the functionality of the base dependent hardware, firmware and/or software that is relied upon by the dependent component TSF.

1681 The evaluator assesses the description of the security functionality that the dependent component TSF requires to be provided by the base component's hardware, firmware and software. The emphasis of this work unit is on the

level of detail of this description, rather than on an assessment of the information's accuracy. (The assessment of the accuracy of the information is the focus of the next work unit.)

1682 This description of the base component's functionality need not be any more detailed than the level of the description of a component of the TSF, as would be provided in the TOE Design (TOE design (ADV_TDS))

ACO_REL.1-2 The evaluator ***shall examine*** the reliance information to determine that it accurately reflects the objectives specified for the operational environment of the dependent component.

1683 The reliance information contains the description of the base component's security functionality relied upon by the dependent component. To ensure that the reliance information is consistent with the expectations of the operational environment of the dependent component, the evaluator compares the reliance information with the statement of objectives for the environment in the ST for the dependent component.

1684 For example, if the reliance information claims that the dependent component TSF relies upon the base component to store and protect audit data, yet other evaluation evidence (e.g. the dependent component design) makes it clear that the dependent component TSF itself is storing and protecting the audit data, this would indicate an inaccuracy.

1685 It should be noted that the objectives for the operational environment may include objectives that can be met by non-IT measures. While the services that the base component environment is expected to provide may be described in the description of IT objectives for the operational environment in the dependent component ST, it is not required that all such expectations on the environment be described in the reliance information.

ACO_REL.1.2C ***The reliance information shall describe all interactions through which the dependent component TSF requests services from the base component.***

ACO_REL.1-3 The evaluator ***shall examine*** the reliance information to determine that it describes all interactions between the dependent component and the base component, through which the dependent component TSF requests services from the base component.

1686 The dependent component TSF may request services of the base component that were not within the TSF of the base component (see B.3, Interactions between composed IT entities in CC Part 3).

1687 The interfaces to the base component's functionality are described at the same level as the description of the interfaces to the dependent component TSF functionality, as would be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)).

1688 The purpose of describing the interactions between the dependent component and the base component is to provide an understanding of how the dependent

component TSF relies upon the base component for the provision of services to support the operation of security functionality of the dependent component. These interactions do not need to be characterised at the implementation level (e.g. parameters passed from one routine in a component to a routine in another component), but the data elements identified for a particular component that are going to be used by another component should be covered in this description. The statement should help the reader understand in general why the interaction is necessary.

1689 Accuracy and completeness of the interfaces is based on the security functionality that the TSF requires to be provided by the base component, as assessed in work units ACO_REL.1-1 and ACO_REL.1-2. It should be possible to map all of the functionality described in the earlier work units to the interfaces identified in this work unit, and vice versa. An interface that does not correspond to described functionality would also indicate an inadequacy.

ACO_REL.1.3C ***The reliance information shall describe how the dependent TSF protects itself from interference and tampering by the base component.***

ACO_REL.1-4 The evaluator ***shall examine*** the reliance information to determine that it describes how the dependent TSF protects itself from interference and tampering by the base component describes the method of use for each interface.

1690 The description of how the dependent component protects itself from interference and tampering by the base component is to be provided at the same level of detail as necessary for ADV_ARC.1-4.

16.5.2 Evaluation of sub-activity (ACO_REL.2)

16.5.2.1 Objectives

1691 The objectives of this sub-activity are to determine whether the developer's reliance evidence provides sufficient information to determine that the necessary functionality is available in the base component, and the means by which that functionality is invoked. This is provided in terms of the interfaces between the dependent and base component and the return values from those interfaces called by the dependent component.

16.5.2.2 Input

1692 The evaluation evidence for this sub-activity is:

a) the composed ST;

b) the dependent component functional specification;

c) the dependent component design;

d) the dependent component implementation representation;

e) the dependent component architectural design;

f) the reliance information.

16.5.2.3 Application notes

1693 A dependent component whose TSF interacts with the base component requires functionality provided by that base component (e.g., remote authentication, remote audit data storage). In these cases, those invoked services need to be described for those charged with configuring the composed TOE for end users. The rationale for requiring this documentation is to aid integrators of the composed TOE to determine what services in the base component might have adverse effects on the dependent component, and to provide information against which to determine the compatibility of the components when applying the Development evidence (ACO_DEV) family.

16.5.2.4 Action ACO_REL.2.1E

ACO_REL.2.1C ***The reliance information shall describe the functionality of the base component hardware, firmware and/or software that is relied upon by the dependent component TSF.***

ACO_REL.2-1 The evaluator ***shall check*** the reliance information to determine that it describes the functionality of the base dependent hardware, firmware and/or software that is relied upon by the dependent component TSF.

1694 The evaluator assesses the description of the security functionality that the dependent component TSF requires to be provided by the base component's hardware, firmware and software. The emphasis of this work unit is on the level of detail of this description, rather than on an assessment of the information's accuracy. (The assessment of the accuracy of the information is the focus of the next work unit.)

1695 This description of the base component's functionality need not be any more detailed than the level of the description of a component of the TSF, as would be provided in the TOE Design (TOE design (ADV_TDS))

ACO_REL.2-2 The evaluator ***shall examine*** the reliance information to determine that it accurately reflects the objectives specified for the operational environment of the dependent component.

1696 The reliance information contains the description of the base component's security functionality relied upon by the dependent component. To ensure that the reliance information is consistent with the expectations of the operational environment of the dependent component, the evaluator compares the reliance information with the statement of objectives for the environment in the ST for the dependent component.

1697 For example, if the reliance information claims that the dependent component TSF relies upon the base component to store and protect audit

data, yet other evaluation evidence (e.g. the dependent component design) makes it clear that the dependent component TSF itself is storing and protecting the audit data, this would indicate an inaccuracy.

1698 It should be noted that the objectives for the operational environment may include objectives that can be met by non-IT measures. While the services that the base component environment is expected to provide may be described in the description of IT objectives for the operational environment in the dependent component ST, it is not required that all such expectations on the environment be described in the reliance information.

ACO_REL.2.2C ***The reliance information shall describe all interactions through which the dependent component TSF requests services from the base component.***

ACO_REL.2-3 The evaluator ***shall examine*** the reliance information to determine that it describes all interactions between the dependent component and the base component, through which the dependent component TSF requests services from the base component.

1699 The dependent component TSF may request services of the base component that were not within the TSF of the base component (see Annex B.3, Interactions between composed IT entities in CC Part 3).

1700 The interfaces to the base component's functionality are described at the same level as the description of the interfaces to the dependent component TSF functionality, as would be provided between subsystems in the TOE design (Evaluation of sub-activity (ADV_TDS.1)).

1701 The purpose of describing the interactions between the dependent component and the base component is to provide an understanding of how the dependent component TSF relies upon the base component for the provision of services to support the operation of security functionality of the dependent component. These interactions do not need to be characterised at the implementation level (e.g. parameters passed from one routine in a component to a routine in another component), but the data elements identified for a particular component that are going to be used by another component should be covered in this description. The statement should help the reader understand in general why the interaction is necessary.

1702 Accuracy and completeness of the interfaces is based on the security functionality that the TSF requires to be provided by the base component, as assessed in work units ACO_REL.2-1 and ACO_REL.2-2. It should be possible to map all of the functionality described in the earlier work units to the interfaces identified in this work unit, and vice versa. An interface that does not correspond to described functionality would also indicate an inadequacy.

ACO_REL.2.3C ***The reliance information shall describe each interaction in terms of the interface used and the return values from those interfaces.***

ACO_REL.2-4 The reliance information shall describe each interaction in terms of the interface used and the return values from those interfaces.

1703 The identification of the interfaces used by the dependent component TSF when making services requests of the base component allows an integrator to determine whether the base component provides all the necessary corresponding interfaces. This understanding is further gained through the specification of the return values expected by the dependent component. The evaluator ensures that interfaces are described for each interaction specified (as analysed in ACO_REL.2-3).

ACO_REL.2.4C ***The reliance information shall describe how the dependent TSF protects itself from interference and tampering by the base component.***

ACO_REL.2-5 The evaluator ***shall examine*** the reliance information to determine that it describes how the dependent TSF protects itself from interference and tampering by the base component describes the method of use for each interface.

1704 The description of how the dependent component protects itself from interference and tampering by the base component is to be provided at the same level of detail as necessary for ADV_ARC.1-4.

16.6 Composed TOE testing (ACO_CTT)

16.6.1 Evaluation of sub-activity (ACO_CTT.1)

16.6.1.1 Objectives

1705 The objective of this sub-activity is to determine whether the developer correctly performed and documented tests for each of the base component interfaces on which the dependent component relies. As part of this determination the evaluator repeats a sample of the tests performed by the developer and performs any additional tests required to ensure the expected behaviour of all composed TOE SFRs and interfaces of the base component relied upon by the dependent component is demonstrated.

16.6.1.2 Input

1706 The evaluation evidence for this sub-activity is:

a) the composed TOE suitable for testing;

b) the composed TOE testing evidence;

c) the reliance information;

d) the development information.

16.6.1.3 Action ACO_CTT.1.1E

ACO_CTT.1.1C ***The composed TOE and base component interface test documentation shall consist of test plans, expected test results and actual test results.***

ACO_CTT.1-1 The evaluator ***shall examine*** the composed TOE test documentation to determine that it consists of test plans, expected test results and actual test results.

1707 This work unit may be satisfied by provision of the test evidence from the evaluation of the dependent component if the base component was used to satisfy the requirements for IT in the operational environment of the dependent component.

1708 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

a) that the test documentation consist of test plans expected test results and actual test results;

b) that the test documentation contains the information necessary to ensure the tests are repeatable;

c) the level of developer effort that was applied to testing of the base component.

ACO_CTT.1-2 The evaluator ***shall examine*** the base component interface test documentation to determine that it consists of test plans, expected test results and actual test results.

1709 This work unit may be satisfied by provision of the test evidence from the evaluation of the base component for those interfaces relied upon in the composed TOE by the dependent component are TSFIs of the successfully evaluated base component. The determination of whether the interfaces of the base component relied upon by the dependent component were in fact TSFIs of the evaluated base component is made during the ACO_COR activity.

1710 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

a) that the test documentation consist of test plans expected test results and actual test results;

b) that the test documentation contains the information necessary to ensure the tests are repeatable;

c) the level of developer effort that was applied to testing of the base component.

ACO_CTT.1.2C ***The test documentation from the developer execution of the composed TOE tests shall demonstrate that the TSF behaves as specified.***

ACO_CTT.1-3 The evaluator ***shall examine*** the test documentation to determine that the developer execution of the composed TOE tests shall demonstrate that the TSF behaves as specified.

1711 The evaluator should construct a mapping between the tests described in the test plan and the SFRs specified for the composed TOE to identify which SFRs have been tested by the developer.

1712 Guidance on this work unit can be found in:

a) Chapter 14.2.1.

b) Chapter 14.2.2.

1713 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be compared with the mapping to determine that the SFRs of the composed TOE, as tested by the developer, behave as expected.

ACO_CTT.1.3C ***The test documentation from the developer execution of the base component interface tests shall demonstrate that the base component interface relied upon by the dependent component behaves as specified.***

ACO_CTT.1-4 The evaluator ***shall examine*** the test documentation to determine that the developer execution of the base component interface tests shall demonstrate

that the base component interfaces relied upon by the dependent component behave as specified.

1714 The evaluator should construct a mapping between the tests described in the test plan and the interfaces of the base component relied upon by the dependent component (as specified in the reliance information, examined under ACO_REL) to identify which base component interfaces have been tested by the developer.

1715 Guidance on this work unit can be found in:

a) Chapter 14.2.1.

b) Chapter 14.2.2.

1716 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be compared with the mapping to determine that the interfaces of the base component, as tested by the developer, behave as expected.

ACO_CTT.1.4C ***The base component shall be suitable for testing.***

ACO_CTT.1-5 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly and is in a known state.

1717 To determine that the composed TOE has been installed properly and is in a known state the ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the TOE provided by the developer for testing.

ACO_CTT.1-6 The evaluator ***shall examine*** the set of resources provided by the developer to determine that they are equivalent to the set of resources used by the base component developer to functionally test the base component.

1718 To determine that the set of resources provided are equivalent to those used to functionally test the base component as used in the composed TOE, the ATE_IND.2-3 work unit will be applied.

16.6.1.4 Action ACO_CTT.1.2E

ACO_CTT.1-7 The tests are to be selected and executed in accordance with ATE_IND.2.2E, to demonstrate the correct behaviour of the SFRs specified in the composed TOE security target.

1719 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.2E, reporting in the ETR for the composed TOE all analysis, results and verdicts as dictated by the associated work units.

16.6.1.5 Action ACO_CTT.1.3E

ACO_CTT.1-8 The evaluator ***shall perform*** testing, in accordance with ATE_IND.2.3E, for a subset of the SFRs specified in the composed TOE security target to operate as specified.

1720 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

1721 When selecting interfaces of the TSF of the composed TOE to test, the evaluator should take into account any modifications to the components from the evaluated version or configuration. Modifications to the component from that evaluated may include patches introduced, a different configuration as a result of modified guidance documentation, reliance an additional portion of the component that was not within the TSF of the component. These modifications will have been identified during the Composition rationale (ACO_COR) activity.

16.6.2 Evaluation of sub-activity (ACO_CTT.2)

16.6.2.1 Objectives

1722 The objective of this sub-activity is to determine whether the developer correctly performed and documented tests for each of the base component interfaces on which the dependent component relies. As part of this determination the evaluator repeats a sample of the tests performed by the developer and performs any additional tests required to fully demonstrate the expected behaviour of the composed TOE and the interfaces of the base component relied upon by the dependent component.

16.6.2.2 Input

1723 The evaluation evidence for this sub-activity is:

a) the composed TOE suitable for testing;

b) the composed TOE testing evidence;

c) the reliance information;

d) the development information.

16.6.2.3 Action ACO_CTT.2.1E

ACO_CTT.2.1C ***The composed TOE and base component interface test documentation shall consist of test plans, expected test results and actual test results.***

ACO_CTT.2-1 The evaluator ***shall examine*** the composed TOE test documentation to determine that it consists of test plans, expected test results and actual test results.

1724 This work unit may be satisfied by provision of the test evidence from the evaluation of the dependent component if the base component was used to satisfy the requirements for IT in the operational environment of the dependent component.

1725 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

a) that the test documentation consist of test plans expected test results and actual test results;

b) that the test documentation contains the information necessary to ensure the tests are repeatable;

c) the level of developer effort that was applied to testing of the base component.

ACO_CTT.2-2 The evaluator ***shall examine*** the base component interface test documentation to determine that it consists of test plans, expected test results and actual test results.

1726 This work unit may be satisfied by provision of the test evidence from the evaluation of the base component for those interfaces relied upon in the composed TOE by the dependent component are TSFIs of the successfully evaluated base component. The determination of whether the interfaces of the base component relied upon by the dependent component were in fact TSFIs of the evaluated base component is made during the ACO_COR activity.

1727 All work units necessary for the satisfaction of ATE_FUN.1.1E will be applied to determine:

a) that the test documentation consist of test plans expected test results and actual test results;

b) that the test documentation contains the information necessary to ensure the tests are repeatable;

c) the level of developer effort that was applied to testing of the base component.

ACO_CTT.2.2C ***The test documentation from the developer execution of the composed TOE tests shall demonstrate that the TSF behaves as specified and is complete.***

ACO_CTT.2-3 The evaluator ***shall examine*** the test documentation to determine that it provides accurate correspondence between the tests in the test documentation relating to the testing of the composed TOE and the composed TOE SFRs in the composed TOE security target.

1728 A simple cross-table may be sufficient to show test correspondence. The identification of correspondence between the tests and SFRs presented in the test documentation has to be unambiguous.

ACO_CTT.2-4 The evaluator ***shall examine*** the test documentation to determine that the developer execution of the composed TOE tests shall demonstrate that the TSF behaves as specified.

1729 Guidance on this work unit can be found in:

a) Chapter 14.2.1.

b) Chapter 14.2.2.

1730 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be compared with the mapping to determine that the SFRs of the composed TOE, as tested by the developer, behave as expected.

ACO_CTT.2.3C ***The test documentation from the developer execution of the base component interface tests shall demonstrate that the base component interface relied upon by the dependent component behaves as specified and is complete.***

ACO_CTT.2-5 The evaluator ***shall examine*** the test documentation to determine that it provides accurate correspondence between the tests in the test documentation relating to the testing of the base component interfaces relied upon by the dependent component and the interfaces specified in the reliance information.

1731 A simple cross-table may be sufficient to show test correspondence. The identification of correspondence between the tests and interfaces presented in the test documentation has to be unambiguous.

ACO_CTT.2-6 The evaluator ***shall examine*** the test documentation to determine that the developer execution of the base component interface tests shall demonstrate that the base component interfaces relied upon by the dependent component behave as specified.

1732 Guidance on this work unit can be found in:

a) Chapter 14.2.1.

b) Chapter 14.2.2.

1733 The outputs from the successful execution of the tests (as assessed for ATE_FUN.1.3C can be compared with the mapping to determine that the interfaces of the base component, as tested by the developer, behave as expected.

ACO_CTT.2.4C ***The base component shall be suitable for testing.***

ACO_CTT.2-7 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly and is in a known state.

1734 To determine that the composed TOE has been installed properly and is in a known state the ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the TOE provided by the developer for testing.

ACO_CTT.2-8 The evaluator ***shall examine*** the set of resources provided by the developer to determine that they are equivalent to the set of resources used by the base component developer to functionally test the base component.

1735 To determine that the set of resources provided are equivalent to those used to functionally test the base component as used in the composed TOE, the ATE_IND.2-3 work unit will be applied.

16.6.2.4 Action ACO_CTT.2.2E

ACO_CTT.2-9 The tests are to be selected and executed in accordance with ATE_IND.2.2E, to demonstrate the correct behaviour of the SFRs specified in the composed TOE security target.

1736 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.2E, reporting in the ETR for the composed TOE all analysis, results and verdicts as dictated by the associated work units.

16.6.2.5 Action ACO_CTT.2.3E

ACO_CTT.2-10 The evaluator ***shall perform*** testing, in accordance with ATE_IND.2.3E, for a subset of the SFRs specified in the composed TOE security target to operate as specified.

1737 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

1738 When selecting interfaces of the TSF of the composed TOE to test, the evaluator should take into account any modifications to the components from the evaluated version or configuration. Modifications to the component from that evaluated may include patches introduced, a different configuration as a result of modified guidance documentation, reliance an additional portion of the component that was not within the TSF of the component. These modifications will have been identified during the Composition rationale (ACO_COR) activity.

ACO_CTT.2-11 The evaluator ***shall perform*** testing, in accordance with Evaluation of sub-activity (ATE_IND.2), for a subset of the interfaces to the base component to confirm they operate as specified.

1739 The evaluator will apply all work units necessary for the satisfaction of ATE_IND.2.3E, reporting in the ETR for the composed TOE all analysis, results and verdicts as dictated by the work units.

1740 When selecting interfaces of the base component to test, the evaluator should take into account any modifications to the base component from the evaluated version or configuration. In particular, the evaluator should consider the development of tests to demonstrate the correct behaviour of interfaces of the base component that were not considered during the evaluation of the base component. These additional interfaces and other

modifications to the base component will have been identified during the Composition rationale (ACO_COR) activity.

16.7 Composition vulnerability analysis (ACO_VUL)

16.7.1 Evaluation of sub-activity (ACO_VUL.1)

16.7.1.1 Objectives

1741 The objective of this sub-activity is to determine whether the composed TOE, in its operational environment, has easily exploitable vulnerabilities.

1742 The developer provides details of any residual vulnerabilities reported from evaluation of the components. The evaluator performs an analysis of the disposition the residual vulnerabilities reported and also performs a search of the public domain, to identify any new potential vulnerabilities in the components (i.e. those issues that have been reported in the public domain since evaluation of the base component). The evaluator then performs penetration testing to demonstrate that the potential vulnerabilities cannot be exploited in the TOE, in its operational environment, by an attacker with basic attack potential.

16.7.1.2 Input

1743 The evaluation evidence for this sub-activity is:

a) the composed TOE suitable for testing;

b) the composed ST;

c) the composition rationale;

d) the guidance documentation;

e) information publicly available to support the identification of possible security vulnerabilities;

f) residual vulnerabilities reported during evaluation of each component.

16.7.1.3 Application notes

1744 See the application notes for Evaluation of sub-activity (AVA_VAN.1).

16.7.1.4 Action ACO_VUL.1.1E

ACO_VUL.1.1C ***The composed TOE shall be suitable for testing.***

ACO_VUL.1-1 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly and is in a known state.

1745 To determine that the composed TOE has been installed properly and is in a known state the ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.

1746 If the assurance package includes a component from the ACO_CTT family, then the evaluator may refer to the result of the work unit ACO_CTT*-1 to demonstrate this has been satisfied.

ACO_VUL.1-2 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions and objectives in the STs the components relating to IT entities for are fulfilled by the other components.

1747 The STs for the component may include assumptions about other components that may use the component to which the ST relates, e.g. the ST for an operating system used as a base component may include an assumption that any applications loaded on the operating system do not run in privileged mode. These assumptions and objectives are to be fulfilled by other components in the composed TOE.

16.7.1.5 Action ACO_VUL.1.2E

ACO_VUL.1-3 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1748 The list of vulnerabilities identified in the product during the evaluation of the base component, which were demonstrated to be non-exploitable in the base component, is to be used as an input into this activity. The evaluator will determine that the premise(s) on which a vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the base component it was assumed that a particular operating system service was disabled, which is enabled in the composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped out should now be considered.

1749 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. dependent component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

ACO_VUL.1-4 The evaluator ***shall examine*** the residual vulnerabilities from the dependent component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1750 The list of vulnerabilities identified in the product during the evaluation of the dependent component, which were demonstrated to be non-exploitable in the dependent component, is to be used as an input into this activity. The evaluator will determine that the premise(s) on which a vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the dependent component it was assumed that IT

meeting the operational environment requirements would not return a certain value in response to a service request, which is provided by the base component in the composed TOE evaluation, any potential vulnerabilities relating to that return value previously scoped out should now be considered.

1751 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the dependent component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. base component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

16.7.1.6 Action ACO_VUL.1.3E

ACO_VUL.1-5 The evaluator ***shall examine*** the sources of information publicly available to support the identification of possible security vulnerabilities in the base component that have become known since the completion of evaluation of the base component.

1752 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search for vulnerabilities in the base component.

1753 Those potential vulnerabilities that were publicly available prior to the evaluation of the base component do not have to be further investigated unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since the base component evaluation that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.1-6 The evaluator ***shall examine*** the sources of information publicly available to support the identification of possible security vulnerabilities in the dependent component that have become known since the completion of the dependent component evaluation.

1754 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search for vulnerabilities in the dependent component.

1755 Those potential vulnerabilities that were publicly available prior to the evaluation of the dependent component do not have to be further investigated unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since evaluation of the dependent component that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.1-7 The evaluator ***shall record*** in the ETR the identified potential security vulnerabilities that are candidates for testing and applicable to the composed TOE in its operational environment.

1756 The ST, guidance documentation and functional specification are used to determine whether the vulnerabilities are relevant to the composed TOE in its operational environment.

1757 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the evaluator determines that the vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1758 A list of potential vulnerabilities applicable to the composed TOE in its operational environment, which can be used as an input into penetration testing activities (i.e. ACO_VUL.1.4E), shall be reported in the ETR by the evaluators.

16.7.1.7 Action ACO_VUL.1.4E

ACO_VUL.1-8 The evaluator ***shall conduct*** penetration testing as detailed for AVA_VAN.1.3E.

1759 The evaluator will apply all work units necessary for the satisfaction of evaluator action AVA_VAN.1.3E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by the work units.

1760 The evaluator will also apply the work units for the evaluator action AVA_VAN.1.1E to determine that the composed TOE provided by the developer is suitable for testing.

16.7.2 Evaluation of sub-activity (ACO_VUL.2)

16.7.2.1 Objectives

1761 The objective of this sub-activity is to determine whether the composed TOE, in its operational environment, has vulnerabilities exploitable by attackers possessing basic attack potential.

1762 The developer provides an analysis of the disposition of any residual vulnerabilities reported for the components and of any vulnerabilities introduced through the combination of the base and dependent components. The evaluator performs a search of the public domain to identify any new potential vulnerabilities in the components (i.e. those issues that have been reported in the public domain since the completion of the evaluation of the components). The evaluator will also perform an independent vulnerability analysis of the composed TOE and penetration testing.

16.7.2.2 Input

1763 The evaluation evidence for this sub-activity is:

a) the composed TOE suitable for testing;

b) the composed ST;

c) the composition rationale;

d) the reliance information;

e) the guidance documentation;

f) information publicly available to support the identification of possible security vulnerabilities.

g) residual vulnerabilities reported during evaluation of each component.

16.7.2.3 Application notes

1764 See the application notes for Evaluation of sub-activity (AVA_VAN.2).

16.7.2.4 Action ACO_VUL.2.1E

ACO_VUL.2.1C ***The composed TOE shall be suitable for testing.***

ACO_VUL.2-1 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly and is in a known state.

1765 To determine that the composed TOE has been installed properly and is in a known state the ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.

1766 If the assurance package includes ACO_CTT family, then the evaluator may refer to the result of the work unit Composed TOE testing (ACO_CTT)*-1 to demonstrate this has been satisfied.

ACO_VUL.2-2 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions and objectives in the STs the components relating to IT entities for are fulfilled by the other components.

1767 The STs for the component may include assumptions about other components that may use the component to which the ST relates, e.g. the ST for an operating system used as a base component may include an assumption that any applications loaded on the operating system do not run in privileged mode. These assumptions and objectives are to be fulfilled by other components in the composed TOE.

16.7.2.5 Action ACO_VUL.2.2E

ACO_VUL.2-3 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1768 The list of vulnerabilities identified in the product during the evaluation of the base component, which were demonstrated to be non-exploitable in the base component, is to be used as an input into this activity. The evaluator will determine that the premise(s) on which a vulnerability was deemed to be

non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the base component it was assumed that a particular operating system service was disabled, which is enabled in the composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped out should now be considered.

1769 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. dependent component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

ACO_VUL.2-4 The evaluator ***shall examine*** the residual vulnerabilities from the dependent component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1770 The list of vulnerabilities identified in the product during the evaluation of the dependent component, which were demonstrated to be non-exploitable in the dependent component, is to be used as an input into this activity. The evaluator will determine that the premise(s) on which a vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the dependent component it was assumed that IT meeting the operational environment requirements would not return a certain value in response to a service request, which is provided by the base component in the composed TOE evaluation, any potential vulnerabilities relating to that return value previously scoped out should now be considered.

1771 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the dependent component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. base component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

16.7.2.6 Action ACO_VUL.2.3E

ACO_VUL.2-5 The evaluator examines the sources of information publicly available to support the identification of possible security vulnerabilities in the base component that have become known since the completion of the base component evaluation.

1772 The evaluator will use the information in the public domain as described in AVA_VAN.2-2 to search for vulnerabilities in the base component.

1773 Those potential vulnerabilities that were publicly available prior to the evaluation of the base component do not have to be further investigated

unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since the base component evaluation that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.2-6 The evaluator ***shall examine*** the sources of information publicly available to support the identification of possible security vulnerabilities in the dependent component that have become known since the completion of the dependent component evaluation.

1774 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search for vulnerabilities in the dependent component.

1775 Those potential vulnerabilities that were publicly available prior to the evaluation of the dependent component do not have to be further investigated unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since evaluation of the dependent component that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.2-7 The evaluator ***shall record*** in the ETR the identified potential security vulnerabilities that are candidates for testing and applicable to the composed TOE in its operational environment.

1776 The ST, guidance documentation and functional specification are used to determine whether the vulnerabilities are relevant to the composed TOE in its operational environment.

1777 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the evaluator determines that the vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1778 A list of potential vulnerabilities applicable to the composed TOE in its operational environment, which can be used as an input into penetration testing activities (ACO_VUL.2.5E), shall be reported in the ETR by the evaluators.

16.7.2.7 Action ACO_VUL.2.4E

ACO_VUL.2-8 The evaluator ***shall conduct*** a search of the composed TOE ST, guidance documentation, reliance information and composition rationale to identify possible security vulnerabilities in the composed TOE.

1779 The consideration of the components of the composed TOE in the independent evaluator vulnerability analysis will take a slightly different form to that documented in AVA_VAN.2.3E for a component evaluation, as it will not necessarily consider all layers of design abstraction relevant to the assurance package. These will have already been considered during the

evaluation of the components, but the evidence may not be available for the composed TOE evaluation. However, the general approach described in the work units associated with AVA_VAN.2.3E is applicable and should form the basis of the evaluator's search for potential vulnerabilities in the composed TOE.

1780 A vulnerability analysis of the individual components used in the composed TOE will have already been performed during evaluation of the individual components. The focus of the vulnerability analysis during the composed TOE evaluation is to identify any vulnerabilities introduced as a result of the integration of the components or due to any changes in the use of the components between the evaluated component configuration to the composed TOE configuration.

1781 The evaluator will use the understanding of the component's construction as detailed in the reliance information for the dependent component, and the development information and composition rationale for the base component, together with the dependent component design information. This information will allow the evaluator to gain an understanding of how the base component and dependent component interact and identify potential vulnerabilities that may be introduced as a result of this interaction.

1782 The evaluator will consider any new guidance provided for the installation, start-up and operation of the composed TOE to identify any potential vulnerabilities introduced through this revised guidance.

1783 If any of the individual components have been through assurance continuity activities since the completion of the component evaluation, the evaluator will consider the patch(es) in the independent vulnerability analysis. Information related to the change provided in a public report of the assurance continuity activities (e.g. Maintenance Report) will be the main source of input material of the change. This will be supplemented by any updates to the guidance documentation resulting from the change and any information regarding the change available in the public domain, e.g. vendor website.

1784 Any risks identified due to the lack of evidence to establish the full impact of any patches or deviations in the configuration of a component from the evaluated configuration are to be documented in the evaluator's vulnerability analysis.

16.7.2.8 Action ACO_VUL.2.5E

ACO_VUL.2-9 The evaluator ***shall conduct*** penetration testing as detailed for AVA_VAN.2.4E.

1785 The evaluator will apply all work units necessary for the satisfaction of evaluator action AVA_VAN.2.4E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by the work units.

1786 The evaluator will also apply the work units for the evaluator action AVA_VAN.2.1E to determine that the composed TOE provided by the developer is suitable for testing.

16.7.3 Evaluation of sub-activity (ACO_VUL.3)

16.7.3.1 Objectives

1787 The objective of this sub-activity is to determine whether the composed TOE, in its operational environment, has vulnerabilities exploitable by attackers possessing extended-basic attack potential.

1788 The developer provides an analysis of the disposition of any residual vulnerabilities reported for the components and of any vulnerabilities introduced through the combination of the base and dependent components. The evaluator performs a search of the public domain to identify any new potential vulnerabilities in the components (i.e. those issues that have been reported in the public domain since the completion of the component evaluations). The evaluator will also perform an independent vulnerability analysis of the composed TOE and penetration testing.

16.7.3.2 Input

1789 The evaluation evidence for this sub-activity is:

a) the composed TOE suitable for testing;

b) the composed ST;

c) the composition rationale;

d) the reliance information;

e) the guidance documentation;

f) information publicly available to support the identification of possible security vulnerabilities.

g) residual vulnerabilities reported during evaluation of each component.

16.7.3.3 Application notes

1790 See the application notes for Evaluation of sub-activity (AVA_VAN.3).

16.7.3.4 Action ACO_VUL.3.1E

ACO_VUL.3.1C ***The composed TOE shall be suitable for testing.***

ACO_VUL.3-1 The evaluator ***shall examine*** the composed TOE to determine that it has been installed properly and is in a known state.

1791 To determine that the composed TOE has been installed properly and is in a known state the ATE_IND.2-1 and ATE_IND.2-2 work units will be applied to the composed TOE.

1792 If the assurance package includes ACO_CTT family, then the evaluator may refer to the result of the work unit Composed TOE testing (ACO_CTT)*-1 to demonstrate this has been satisfied.

ACO_VUL.3-2 The evaluator ***shall examine*** the composed TOE configuration to determine that any assumptions and objectives in the STs the components relating to IT entities for are fulfilled by the other components.

1793 The STs for the component may include assumptions about other components that may use the component to which the ST relates, e.g. the ST for an operating system used as a base component may include an assumption that any applications loaded on the operating system do not run in privileged mode. These assumptions and objectives are to be fulfilled by other components in the composed TOE.

16.7.3.5 Action ACO_VUL.3.2E

ACO_VUL.3-3 The evaluator ***shall examine*** the residual vulnerabilities from the base component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1794 The list of vulnerabilities identified in the product during the evaluation of the base component, which were demonstrated to be non-exploitable in the base component, is to be used as an input into this activity. The evaluator will determine that the premise(s) on which a vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the base component it was assumed that a particular operating system service was disabled, which is enabled in the composed TOE evaluation, any potential vulnerabilities relating to that service previously scoped out should now be considered.

1795 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the base component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. dependent component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

ACO_VUL.3-4 The evaluator ***shall examine*** the residual vulnerabilities from the dependent component evaluation to determine that they are not exploitable in the composed TOE in its operational environment.

1796 The list of vulnerabilities identified in the product during the evaluation of the dependent component, which were demonstrated to be non-exploitable in the dependent component, is to be used as an input into this activity. The

evaluator will determine that the premise(s) on which a vulnerability was deemed to be non-exploitable is upheld in the composed TOE, or whether the combination has re-introduced the potential vulnerability. For example, if during evaluation of the dependent component it was assumed that IT meeting the operational environment requirements would not return a certain value in response to a service request, which is provided by the base component in the composed TOE evaluation, any potential vulnerabilities relating to that return value previously scoped out should now be considered.

1797 Also, this list of known, non-exploitable vulnerabilities resulting from the evaluation of the dependent component should be considered in the light of any known, non-exploitable vulnerabilities for the other components (e.g. base component) within the composed TOE. This is to consider the case where a potential vulnerability that is non-exploitable in isolation is exploitable when integrated with an IT entity containing another potential vulnerability.

16.7.3.6 Action ACO_VUL.3.3E

ACO_VUL.3-5 The evaluator examines the sources of information publicly available to support the identification of possible security vulnerabilities in the base component that have become known since the completion of the base component evaluation.

1798 The evaluator will use the information in the public domain as described in AVA_VAN.3-2 to search for vulnerabilities in the base component.

1799 Those potential vulnerabilities that were publicly available prior to the evaluation of the base component do not have to be further investigated unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since the base component evaluation that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.3-6 The evaluator ***shall examine*** the sources of information publicly available to support the identification of possible security vulnerabilities in the dependent component that have become known since completion of the dependent component evaluation.

1800 The evaluator will use the information in the public domain as described in AVA_VAN.1-2 to search for vulnerabilities in the dependent component.

1801 Those potential vulnerabilities that were publicly available prior to the evaluation of the dependent component do not have to be further investigated unless it is apparent to the evaluator that the attack potential required by an attacker to exploit the potential vulnerability has been significantly reduced. This may be through the introduction of some new technology since evaluation of the dependent component that means the exploitation of the potential vulnerability has been simplified.

ACO_VUL.3-7 The evaluator ***shall record*** in the ETR the identified potential security vulnerabilities that are candidates for testing and applicable to the composed TOE in its operational environment.

1802 The ST, guidance documentation and functional specification are used to determine whether the vulnerabilities are relevant to the composed TOE in its operational environment.

1803 The evaluator records any reasons for exclusion of vulnerabilities from further consideration if the evaluator determines that the vulnerability is not applicable in the operational environment. Otherwise the evaluator records the potential vulnerability for further consideration.

1804 A list of potential vulnerabilities applicable to the composed TOE in its operational environment, which can be used as an input into penetration testing activities (ACO_VUL.3.5E), shall be reported in the ETR by the evaluators.

16.7.3.7 Action ACO_VUL.3.4E

ACO_VUL.3-8 The evaluator ***shall conduct*** a search of the composed TOE ST, guidance documentation, reliance information and composition rationale to identify possible security vulnerabilities in the composed TOE.

1805 The consideration of the components in the independent evaluator vulnerability analysis will take a slightly different form to that documented in AVA_VAN.3.3E for a component evaluation, as it will not necessarily consider all layers of design abstraction relevant to the assurance package. These will have already been considered during the evaluation of the base component, but the evidence may not be available for the composed TOE evaluation. However, the general approach described in the work units associated with AVA_VAN.3.3E is applicable and should form the basis of the evaluator's search for potential vulnerabilities in the composed TOE.

1806 A vulnerability analysis of the individual components used in the composed TOE will have already been performed during evaluation of the components. The focus of the vulnerability analysis during the composed TOE evaluation is to identify any vulnerabilities introduced as a result of the integration of the components or due to any changes in the use of the components between the configuration of the component determined during the component evaluation and the composed TOE configuration.

1807 The evaluator will use the understanding of the component's construction as detailed in the reliance information for the dependent component, and the composition rationale and development information for the base component, together with the dependent component design information. This information will allow the evaluator to gain an understanding of how the base component and dependent component interact.

1808 The evaluator will consider any new guidance provided for the installation, start-up and operation of the composed TOE to identify any potential vulnerabilities introduced through this revised guidance.

1809 If any of the individual components have been through assurance continuity activities since the completion of the component evaluation, the evaluator will consider the patch in the independent vulnerability analysis. Information related to the change provided in a public report of the assurance continuity activities (e.g. Maintenance Report). This will be supplemented by any updates to the guidance documentation resulting from the change and any information regarding the change available in the public domain, e.g. vendor website.

1810 Any risks identified due to the lack of evidence to establish the full impact of any patches or deviations in the configuration of a component from the evaluated configuration are to be documented in the evaluator's vulnerability analysis.

16.7.3.8 Action ACO_VUL.3.5E

ACO_VUL.3-9 The evaluator ***shall conduct*** penetration testing as detailed for AVA_VAN.3.4E.

1811 The evaluator will apply all work units necessary for the satisfaction of evaluator action AVA_VAN.3.4E, reporting in the ETR for the composed TOE all analysis and verdicts as dictated by the work units.

1812 The evaluator will also apply the work units for the evaluator action AVA_VAN.3.1E to determine that the composed TOE provided by the developer is suitable for testing.

A General evaluation guidance (normative)

A.1 Objectives

1813 The objective of this chapter 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

1814 This Section 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.

1815 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.

1816 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.

1817 Sampling in the cases identified in the CC, and in cases specifically covered in CEM 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 CC requirement that would normally all need to be checked or examined, and where such an action would not be expected to raise assurance correspondingly.

1818 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.

1819 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.

1820 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 Depth (ATE_DPT) criteria) then it would be appropriate for the evaluator to give greater focus to the repetition of developer tests. Note that this does not necessarily imply a requirement for a high percentage sample for repetition of developer tests; indeed, given an extensive developer test set, the evaluator may be able to justify a low percentage sample.

1821 Where the developer has used an automated test suite to perform functional testing, it will usually be easier for the evaluator to re-run the entire test suite rather than repeat only a sample of developer tests. However the evaluator does have an obligation to check that the automatic testing does not give misrepresentative results. The implication is thus that this check must be performed for a sample of the automatic test suite, with the principles for selecting some tests in preference to others and ensuring a sufficient sample size applying equally in this case.

1822 The following principles should be followed whenever sampling is performed:

a) Sampling should not be random, rather it should be chosen such that it is representative of all of the evidence. The sample size and composition must always be justified.

b) When sampling relates to the correct implementation of the TOE, the sample should be representative of all aspects relevant to the areas that are sampled. In particular, the selection should cover a variety of components, interfaces, developer and operational sites (if more than one is involved) and hardware platform types (if more than one is involved). The sample size should be commensurate with the cost effectiveness of the evaluation and will depend on a number of TOE dependent factors (e.g. the size and complexity of the TOE, the amount of documentation).

c) Also, when sampling relates to specifically gaining evidence that the developer testing is repeatable and reproducible the sample used must be sufficient to represent all distinct aspects of developer testing, such as different test regimes. The sample used must be sufficient to detect any systematic problem in the developer's functional testing process. The evaluator contribution resulting from the combination of repeating developer tests and performing independent tests must be sufficient to address the major points of concern for the TOE.

d) Where sampling relates to gaining evidence that a process (e.g. visitor control or design review) the evaluator should sample sufficient information to gain reasonable confidence that the procedure is being followed.

e) The sponsor and developer should not be informed in advance of the exact composition of the sample, subject to ensuring timely delivery of the sample and supporting deliverable, e.g. test harnesses and equipment to the evaluator in accordance with the evaluation schedule.

f) The choice of the sample should be free from bias to the degree possible (one should not always choose the first or last item). Ideally the sample selection should be done by someone other than the evaluator.

1823 Errors found in the sample can be categorised as being either systematic or sporadic. If the error is systematic, the problem should be corrected and a complete new sample taken. If properly explained, sporadic errors might be solved without the need for a new sample, although the explanation should be confirmed. The evaluator should use judgement in determining whether to increase the sample size or use a different sample.

A.3 Dependencies

1824 In general it is possible to perform the required evaluation activities, sub-activities, and actions in any order or in parallel. However, there are different kinds of dependencies which have to be considered by the evaluator. This Section provides general guidance on dependencies between different activities, sub-activities, and actions.

A.3.1 Dependencies between activities

1825 For some cases the different assurance classes may recommend or even require a sequence for the related activities. A specific instance is the ST activity. The ST evaluation activity is started prior to any TOE evaluation activities since the ST provides the basis and context to perform them. However, a final verdict on the ST evaluation may not be possible until the TOE evaluation is complete, since changes to the ST may result from activity findings during the TOE evaluation.

A.3.2 Dependencies between sub-activities

1826 Dependencies identified between components in CC Part 3 have to be considered by the evaluator. Most dependencies are one way, e.g. Evaluation of sub-activity (AVA_VAN.1) claims a dependency on Evaluation of sub-activity (ADV_FSP.1) and Evaluation of sub-activity (AGD_OPE.1). There are also instances of mutual dependencies, where both components depend on each other. An example of this is Evaluation of sub-activity (ATE_FUN.1) and Evaluation of sub-activity (ATE_COV.1).

1827 A sub-activity can be assigned a pass verdict normally only if all those sub-activities are successfully completed on which it has a one-way dependency. For example, a pass verdict on Evaluation of sub-activity (AVA_VAN.1) can normally only be assigned if the sub-activities related to Evaluation of sub-activity (ADV_FSP.1) and Evaluation of sub-activity (AGD_OPE.1) are assigned a pass verdict too. In the case of mutual dependency the ordering of these components is down to the evaluator deciding which sub-activity to perform first. Note this indicates that pass verdicts can normally only be assigned once both sub-activities have been successful.

1828 So when determining whether a sub-activity will impact another sub-activity, the evaluator should consider whether this activity depends on potential evaluation results from any dependent sub-activities. Indeed, it may be the case that a dependent sub-activity will impact this sub-activity, requiring previously completed evaluator actions to be performed again.

1829 A significant dependency effect occurs in the case of evaluator-detected flaws. If a flaw is identified as a result of conducting one sub-activity, the assignment of a pass verdict to a dependent sub-activity may not be possible until all flaws related to the sub-activity upon which it depends are resolved.

A.3.3 Dependencies between actions

1830 It may be the case, that results which are generated by the evaluator during one action are used for performing another action. For example, actions for completeness and consistency cannot be completed until the checks for content and presentation have been completed. This means for example that the evaluator is recommended to evaluate the PP/ST rationale after evaluating the constituent parts of the PP/ST.

A.4 Site Visits

A.4.1 Introduction

1831 The assurance class ALC includes requirements for

a) the application of configuration management, ensuring that the integrity of the TOE is preserved;

b) measures, procedures, and standards concerned with secure delivery of the TOE, ensuring that the security protection offered by the TOE is not compromised during the transfer to the user,

c) security measures, used to protect the development environment.

1832 A development site visit is a useful means whereby the evaluator determines whether procedures are being followed in a manner consistent with that described in the documentation.

1833 Reasons for visiting sites include:

a) to observe the use of the CM system as described in the CM plan;

b) to observe the practical application of delivery procedures as described in the delivery documentation;

c) to observe the application of security measures during development and maintenance of the TOE as described in the development security documentation.

1834 Specific and detailed information is given in work units for those activities where site visits are performed:

a) CM capabilities (ALC_CMC).n with n>=3 (especially work unit ALC_CMC.3-10 = ALC_CMC.4-13 = ALC_CMC.5-19);

b) Delivery (ALC_DEL) (especially work unit ALC_DEL.1-2);

c) Development security (ALC_DVS) (especially work unit ALC_DVS.1-3 = ALC_DVS.2-4).

A.4.2 General Approach

1835 During an evaluation it is often necessary that the evaluator will meet the developer more than once and it is a question of good planning to combine the site visit with another meeting to reduce costs. For example one might combine the site visits for configuration management, for the developer's security and for delivery. It may also be necessary to perform more than one site visit to the same site to allow the checking of all development phases. It should be considered that development could occur at multiple facilities within a single building, multiple buildings at the same site, or at multiple sites.

1836 The first site visit should be scheduled early during the evaluation. In the case of an evaluation which starts during the development phase of the TOE, this will allow corrective actions to be taken, if necessary. In the case of an evaluation which starts after the development of the TOE, an early site visit could allow corrective measures to be put in place if serious deficiencies in the applied procedures emerge. This avoids unnecessary evaluation effort.

1837 Interviews are also a useful means of determining whether the written procedures reflect what is done. In conducting such interviews, the evaluator should aim to gain a deeper understanding of the analysed procedures at the development site, how they are used in practise and whether they are being applied as described in the provided evaluation evidence. Such interviews complement but do not replace the examination of evaluation evidence.

1838 As a first step preparing the site visits the evaluators should perform the evaluator work units concerning the assurance class ALC excluding the aspects describing the results of the site visit. Based on the information provided by the relevant developer documentation and the remaining open questions which were not answered by the documentation the evaluators compile a check list of the questions which are to be resolved by the site visits.

1839 The first version of the evaluation report concerning the ALC class and the check list serves as input for the consultation with the evaluation authority concerning the site visits.

1840 The check list serve as a guide line for the site visits, which questions are to be answered by inspection of the relevant measures, their application and results, and by interviews. Where appropriate, sampling is used for gaining the required level of confidence (see Section A.2).

1841 The results of the site visits are recorded and serve as input for the final version of the evaluation report concerning the assurance class ALC.

1842 Other approaches to gain confidence should be considered that provide an equivalent level of assurance (e.g. to analyse evaluation evidence). Any decision not to make a visit should be determined in consultation with the evaluation authority. Appropriate security criteria and a methodology should be based on other standards of the Information Security Management Systems area.

A.4.3 Orientation Guide for the Preparation of the Check List

1843 In the following some keywords are provided, which topics should be checked during an audit.

A.4.3.1 Aspects of configuration management

1844 Basic

– Items of the configuration list, including TOE, source code, run time libraries, design documentation, development tools (ALC_CMC.3-8).

– Tracking of design documentation, source code, user guidance to different versions of the TOE.

– Integration of the configuration system in the design and development process, test planning, test analysis and quality management procedures.

1845 Test analysis

– Tracking of test plans and results to specific configurations and versions of the TOE.

1846 Access control to development systems

– Policies for access control and logging.

– Policies for project specific assignment and changing of access rights.

1847 Clearance

– Policies for clearance of the TOE and user guidance to the customer.

– Policies for testing and approving of components and the TOE before deployment.

A.4.3.2 Aspects of development security

1848 Infrastructure

– Security measures for physical access control to the development site and rationale for the effectiveness of these measures.

1849 Organisational measures

– Organisational structure of the company in respect of the security of the development environment.

– Organisational separation between development, production, testing and quality assurance.

1850 Personal measures

– Measures for education of the personnel in respect of development security.

– Measures and legal agreements of non disclosure of internal information.

1851 Access control

– Assignment of secured objects (for instance TOE, source code, run time libraries, design documentation, development tools, user guidance) and security policies.

– Policies and responsibilities concerning the access control and the handling of authentication information.

– Policies for logging of any kind access to the development site and protection of the logging data.

1852 Input, processing and output of data

– Security measures for protection of output and output devices (printer, plotter and displays).

– Securing of local networks and communication connections.

1853 Storage, transfer and destruction of documents and data media.

– Policies for handling of documents and data media.

– Policies and responsibilities for destruction of sorted out documents and logging of these events.

1854 Data protection

– Policies and responsibilities for data and information protection (e.g. for performing backups).

1855 Contingency plan

– Practises in case of emergency and responsibilities.

– Documentation of the contingency measures concerning access control.

– Information of the personnel about applicable practises in extreme cases. protection (e.g. for performing backups).

A.4.4 Example of a checklist

1856 The examples of checklists for site visits consist in tables for the preparation of an audit and for the presentation of the results of an audit.

1857 The checklist structure given in the following is preliminary. Dependent on the concrete contents of the new guideline, changes might become necessary.

1858 The checklist is divided into three sections according to the subjects indicated in the introduction (Section A.4.1).

a) Configuration management system.

b) Delivery procedures.

c) Security measures during development.

1859 These sections correspond to the actual CC class ALC, especially the families CM capabilities (ALC_CMC).n with n>=3, Delivery (ALC_DEL) and Development security (ALC_DVS).

1860 The sections are subdivided further into rows corresponding to the relevant work units of the CEM.

1861 The columns of the checklist contain in turn

– a consecutive number,

– the referenced work unit,

– the references to the corresponding developer documentation,

– the explicit reproduction of the developer measures,

– special remarks and questions to be clarified on the visit (beyond the standard evaluator task to verify the application of the indicated measures),

– the result of the examinations during the visit.

1862 If it is decided to have separate checklists for preparation and reporting of the audit, the result column is omitted in the preparation list and the remarks and questions column is omitted in the reporting list. The remaining columns should be identical in both lists.

Table 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 and graded access rights, and of checking identification and authentication of users.	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 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	"Security of the development	The premises are	Is the fencing sufficientl	The evaluators considered

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
	-2	environment", ch. ... (Premises)	protected by security fencing.	y strong and high to prevent an easy intrusion into the premises?	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 not manned. And an access in the goods reception which is secured by two roller shutters.	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 before can be operated only from inside.
...	...	...	...	...	...

A.5 Scheme Responsibilities

1863 This CEM 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 the CEM 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 the CEM should consult with their evaluation schemes to determine under whose auspices the subject falls.)

1864 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 CEM 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;

i) any specific content structure of any written report as a result from an ST evaluation - a scheme may have a specific format for all of its reports detailing results of an evaluation, be it the evaluation of a TOE or of an ST;

j) any additional PP/ST identification information required;

k) any activities to determine the suitability of explicitly-stated requirements in an ST;

l) any requirements for provision of evaluator evidence to support re-evaluation and re-use of evidence;

m) any specific handling of scheme identifiers, logos, trademarks, etc.;

n) any specific guidance in dealing with cryptography;

o) handling and application of scheme, national and international interpretations;

p) a list or characterisations of suitable alternative approaches to testing where testing is infeasible;

q) the mechanism by which an overseer can determine what steps an evaluator took while testing;

r) preferred test approach (if any): at internal interface or at external interface;

s) a list or characterisation of acceptable means of conducting the evaluator's vulnerability analysis (e.g. flaw hypothesis methodology);

t) information regarding any vulnerabilities and weaknesses to be considered.

B Vulnerability Assessment (AVA) (normative)

1865 This annex provides an explanation of the AVA_VAN criteria and examples of their application. This annex does not define the AVA criteria; this definition can be found in CC Part 3 Section Class AVA: Vulnerability assessment.

1866 This annex consists of 2 major parts:

a) *Guidance for completing an independent vulnerability analysis*. This is summarised in section B.1, and described in more detail in section B.2 . These sections describe how an evaluator should approach the construction of an independent Vulnerability Analysis.

b) How to characterise and use assumed Attack Potential of an attacker. This is described in sections B.3 to B.5. These sections provide an example of describe how an attack potential can be characterised and should be used, and provide examples.

B.1 What is Vulnerability Analysis

1867 The purpose of the vulnerability assessment activity is to determine the existence and exploitability of flaws or weaknesses in the TOE in the operational environment. This determination is based upon analysis performed by the evaluator, and is supported by evaluator testing.

1868 At the lowest levels of Vulnerability analysis (AVA_VAN) the evaluator simply performs a search of publicly available information to identify any known weaknesses in the TOE, while at the higher levels the evaluator performs a structured analysis of the TOE evaluation evidence.

1869 There are two main factors in performing a vulnerability analysis, namely;

a) the identification of potential vulnerabilities;

b) penetration testing to determine whether the potential vulnerabilities are exploitable in the operational environment of the TOE.

1870 The identification of vulnerabilities can be further decomposed into the evidence to be searched and how hard to search that evidence to identify potential vulnerabilities. In a similar manner, the penetration testing can be further decomposed into analysis of the potential vulnerability to identify attack methods and the demonstration of the attack methods.

1871 These main factors are iterative in nature, i.e. penetration testing of potential vulnerabilities may lead to the identification of further potential vulnerabilities. Hence, these are performed as a single vulnerability analysis activity.

B.2 Evaluator construction of a Vulnerability Analysis

1872 The evaluator vulnerability analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing a Basic (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for AVA_VAN.3), Moderate (for AVA_VAN.4) or High (for AVA_VAN.5) attack potential. The evaluator first assesses the exploitability of all identified potential vulnerabilities. This is accomplished by conducting penetration testing. The evaluator should assume the role of an attacker with a Basic (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for AVA_VAN.3), Moderate (for AVA_VAN.4) or High (for AVA_VAN.5) attack potential when attempting to penetrate the TOE.

1873 The evaluator considers potential vulnerabilities encountered by the evaluator during the conduct of other evaluation activities. The evaluator penetration testing determining TOE resistance to these potential vulnerabilities should be performed assuming the role of an attacker with a Basic (for AVA_VAN.1 and AVA_VAN.2), Enhanced-Basic (for AVA_VAN.3), Moderate (for AVA_VAN.4) or High (for AVA_VAN.5) attack potential.

1874 However, vulnerability analysis should not be performed as an isolated activity. It is closely linked with ADV and AGD. The evaluator performs these other evaluation activities with a focus on identifying potential vulnerabilities or “areas of concern”. Therefore, evaluator familiarity with the generic vulnerability guidance (provided in Section B.2.1) is required.

B.2.1 Generic vulnerability guidance

1875 The following five categories provide discussion of generic vulnerabilities.

B.2.1.1 Bypassing

1876 Bypassing includes any means by which an attacker could avoid security enforcement, by:

a) exploiting the capabilities of interfaces to the TOE, or of utilities which can interact with the TOE;

b) inheriting privileges or other capabilities that should otherwise be denied;

c) (where confidentiality is a concern) reading sensitive data stored or copied to inadequately protected areas.

1877 Each of the following should be considered (where relevant) in the evaluator's independent vulnerability analysis.

a) Attacks based on exploiting the capabilities of interfaces or utilities generally take advantage of the absence of the required security enforcement on those interfaces. For example, gaining access to

functionality that is implemented at a lower level than that at which access control is enforced. Relevant items include:

1) changing the predefined sequence of invocation of TSFI;

2) invoking an additional TSFI;

3) using a component in an unexpected context or for an unexpected purpose;

4) using implementation detail introduced in less abstract representations;

5) using the delay between time of access check and time of use.

b) Changing the predefined sequence of invocation of components should be considered where there is an expected order in which interfaces to the TOE (e.g. user commands) are called to invoke a TSFI (e.g. opening a file for access and then reading data from it). If a TSFI is invoked through one of the TOE interfaces (e.g. an access control check), the evaluator should consider whether it is possible to bypass the control by performing the call at a later point in the sequence or by missing it out altogether.

c) Executing an additional component (in the predefined sequence) is a similar form of attack to the one described above, but involves the calling of some other TOE interface at some point in the sequence. It can also involve attacks based on interception of sensitive data passed over a network by use of network traffic analysers (the additional component here being the network traffic analyser).

d) Using a component in an unexpected context or for an unexpected purpose includes using an unrelated TOE interface to bypass the TSF by using it to achieve a purpose that it was not designed or intended to achieve. Covert channels are an example of this type of attack (see B.2.1.4 for further discussion of covert channels). The use of undocumented interfaces, which may be insecure, also falls into this category. Such interfaces may include undocumented support and help facilities.

e) Using implementation detail introduced in lower representations may allow an attacker to take advantage of additional functions, resources or attributes that are introduced to the TOE as a consequence of the refinement process. Additional functionality may include test harness code contained in software modules and back-doors introduced during the implementation process.

f) Using the delay between time of check and time of use includes scenarios where an access control check is made and access granted, and an attacker is subsequently able to create conditions in which, had they applied at the time the access check was made, would have

caused the check to fail. An example would be a user creating a background process to read and send highly sensitive data to the user's terminal, and then logging out and logging back in again at a lower sensitivity level. If the background process is not terminated when the user logs off, the MAC checks would have been effectively bypassed.

g) Attacks based on inheriting privileges are generally based on illicitly acquiring the privileges or capabilities of some privileged component, usually by exiting from it in an uncontrolled or unexpected manner. Relevant items include:

1) executing data not intended to be executable, or making it executable;

2) generating unexpected input for a component;

3) invalidating assumptions and properties on which lower-level components rely.

h) Executing data not intended to be executable, or making it executable includes attacks involving viruses (e.g. putting executable code or commands in a file which are automatically executed when the file is edited or accessed, thus inheriting any privileges the owner of the file has).

i) Generating unexpected input for a component can have unexpected effects which an attacker could take advantage of. For example, if the TSF could be bypassed if a user gains access to the underlying operating system, it may be possible to gain such access following the login sequence by exploring the effect of hitting various control or escape sequences whilst a password is being authenticated.

j) Invalidating assumptions and properties on which lower level components rely includes attacks based on breaking out of the constraints of an application to gain access to an underlying operating system in order to bypass the TSF of an application. In this case the assumption being invalidated is that it is not possible for a user of the application to gain such access. A similar attack can be envisaged against an application on an underlying database management system: again the TSF could be bypassed if an attacker can break out of the constraints of the application.

k) Attacks based on reading sensitive data stored in inadequately protected areas (applicable where confidentiality is a concern) include the following issues which should be considered as possible means of gaining access to sensitive data:

1) disk scavenging;

2) access to unprotected memory;

3) exploiting access to shared writable files or other shared resources (e.g. swap files);

4) Activating error recovery to determine what access users can obtain. For example, after a crash an automatic file recovery system may employ a lost and found directory for headerless files, which are on disk without labels. If the TOE implements mandatory access controls, it is important to investigate at what security level this directory is kept (e.g. at system high), and who has access to this directory.

1878 There are a number of different methods through which an evaluator may identify a back-door, including two main techniques. Firstly, by the evaluator inadvertently identifying during testing an interface that can be misused. Secondly, through testing each external interface of the TSF in a debugging mode to identify any modules that are not called as a part of testing the documented interfaces and then inspecting the code that is not called to consider whether it is a back-door.

1879 For a software TOE where Evaluation of sub-activity (ADV_IMP.2) and ALC_TAT.2 or higher components are included in the assurance package, the evaluator may consider during their analysis of the tools the libraries and packages that are linked by the compiler at compilation stage to determine that back-doors are not introduced at this stage.

B.2.1.2 Tampering

1880 Tampering includes any attack based on an attacker attempting to influence the behaviour of the TSF (i.e. corruption or de-activation), for example by:

a) accessing data on whose confidentiality or integrity the TSF relies;

b) forcing the TOE to cope with unusual or unexpected circumstances;

c) disabling or delaying security enforcement;

d) physical modification the TOE.

1881 Each of the following should be considered (where relevant) in the evaluator's independent vulnerability analysis.

a) Attacks based on accessing data, whose confidentiality or integrity are protected, include:

1) reading, writing or modifying internal data directly or indirectly;

2) using a component in an unexpected context or for an unexpected purpose;

3) using interfaces between components that are not visible at a higher level of abstraction.

b) Reading, writing or modifying internal data directly or indirectly includes the following types of attack which should be considered:

1) reading “secrets” stored internally, such as user passwords;

2) spoofing internal data that security enforcing mechanisms rely upon;

3) modifying environment variables (e.g. logical names), or data in configuration files or temporary files.

c) It may be possible to deceive a trusted process into modifying a protected file that it wouldn't normally access.

d) The evaluator should also consider the following “dangerous features”:

1) source code resident on the TOE along with a compiler (for instance, it may be possible to modify the login source code);

2) an interactive debugger and patch facility (for instance, it may be possible to modify the executable image);

3) the possibility of making changes at device controller level, where file protection does not exist;

4) diagnostic code which exists in the source code and that may be optionally included;

5) developer's tools left in the TOE.

e) Using a component in an unexpected context or for an unexpected purpose includes (for example), where the TOE is an application built upon an operating system, users exploiting knowledge of a word processor package or other editor to modify their own command file (e.g. to acquire greater privileges).

f) Using interfaces between components which are not visible at a higher level of abstraction includes attacks exploiting shared access to resources, where modification of a resource by one component can influence the behaviour of another (trusted) component, e.g. at source code level, through the use of global data or indirect mechanisms such as shared memory or semaphores.

g) Attacks based on forcing the TOE to cope with unusual or unexpected circumstances should always be considered. Relevant items include:

1) generating unexpected input for a component;

2) invalidating assumptions and properties on which lower-level components rely.

h) Generating unexpected input for a component includes investigating the behaviour of the TOE when:

1) command input buffers overflow (possibly “crashing the stack” or overwriting other storage, which an attacker may be able to take advantage of, or forcing a crash dump that may contain sensitive information such as clear-text passwords);

2) invalid commands or parameters are entered (including supplying a read-only parameter to an interface which expects to return data via that parameter and supplying improperly formatted input that should fail parsing such as SQL-injection, format strings);

3) an end-of-file marker (e.g. CTRL-Z or CTRL-D) or null character is inserted in an audit trail.

i) Invalidating assumptions and properties on which lower-level components rely includes attacks taking advantage of errors in the source code where the code assumes (explicitly or implicitly) that security relevant data is in a particular format or has a particular range of values. In these cases the evaluator should determine whether they can invalidate such assumptions by causing the data to be in a different format or to have different values, and if so whether this could confer advantage to an attacker.

j) The correct behaviour of the TSF may be dependent on assumptions that are invalidated under extreme circumstances where resource limits are reached or parameters reach their maximum value. The evaluator should consider (where practical) the behaviour of the TOE when these limits are reached, for example:

1) changing dates (e.g. examining how the TOE behaves when a critical date threshold is passed);

2) filling disks;

3) exceeding the maximum number of users;

4) filling the audit log;

5) saturating security alarm queues at a console;

6) overloading various parts of a multi-user TOE which relies heavily upon communications components;

7) swamping a network, or individual hosts, with traffic;

8) filling buffers or fields.

k) Attacks based on disabling or delaying security enforcement include the following items:

1) using interrupts or scheduling functions to disrupt sequencing;

2) disrupting concurrence;

3) using interfaces between components which are not visible at a higher level of abstraction.

l) Using interrupts or scheduling functions to disrupt sequencing includes investigating the behaviour of the TOE when:

1) a command is interrupted (with CTRL-C, CTRL-Y, etc.);

2) a second interrupt is issued before the first is acknowledged.

m) The effects of terminating security critical processes (e.g. an audit daemon) should be explored. Similarly, it may be possible to delay the logging of audit records or the issuing or receipt of alarms such that it is of no use to an administrator (since the attack may already have succeeded).

n) Disrupting concurrence includes investigating the behaviour of the TOE when two or more subjects attempt simultaneous access. It may be that the TOE can cope with the interlocking required when two subjects attempt simultaneous access, but that the behaviour becomes less well defined in the presence of further subjects. For example, a critical security process could be put into a resource-wait state if two other processes are accessing a resource which it requires.

o) Using interfaces between components which are not visible at a higher level of abstraction may provide a means of delaying a time-critical trusted process.

p) Physical attacks can be categorised into physical probing, physical manipulation, physical modification, and substitution.

1) Physical probing by penetrating the TOE targeting internals of the TOE, e.g. reading at internal communication interfaces, lines or memories.

2) Physical manipulation can be with the TOE internals aiming at internal modifications of the TOE (e.g. by using optical fault induction as an interaction process), at the external interfaces of the TOE (e.g. by power or clock glitches) and at the TOE environment (e.g. by modifying temperature).

3) Physical modification of TOE internal security enforcing attributes to inherit privileges or other capabilities that should be denied in regular operation. Such modifications can be caused, e.g., by optical fault induction. Attacks based on physical modification may also yield a modification of the TSF itself, e.g. by causing faults at TOE internal program data transfers before execution. Note, that such kind of bypassing

by modifying the TSF itself can jeopardise every TSF unless there are other measures (possibly environmental measures) that prevent an attacker from gaining physical access to the TOE.

4) Physical substitution to replace the TOE with another IT entity, during delivery or operation of the TOE. Substitution during delivery of the TOE from the development environment to the user should be prevented through application of secure delivery procedures (such as those considered under Development security (ALC_DVS)). Substitution of the TOE during operation may be considered through a combination of user guidance and the operational environment, such that the user is able to be confident that they are interacting with the TOE.

B.2.1.3 Direct attacks

1882 Direct attack includes the identification of any penetration tests necessary to test the strength of permutational or probabilistic mechanism and other mechanisms to ensure they withstand direct attack.

1883 For example, it may be a flawed assumption that a particular implementation of a pseudo-random number generator will possess the required entropy necessary to seed the security mechanism.

1884 Where a probabilistic or permutational mechanism relies on selection of security attribute value (e.g. selection of password length) or entry of data by a human user (e.g. choice of password), the assumptions made should reflect the worst case.

1885 Probabilistic or permutational mechanisms should be identified during examination of evaluation evidence required as input to this sub-activity (security target, functional specification, TOE design and implementation representation subset) and any other TOE (e.g. guidance) documentation may identify additional probabilistic or permutational mechanisms.

1886 Where the design evidence or guidance includes assertions or assumptions (e.g. about how many authentication attempts are possible per minute), the evaluator should independently confirm that these are correct. This may be achieved through testing or through independent analysis.

1887 Direct attacks reliant upon a weakness in a cryptographic algorithm should not be considered under Vulnerability analysis (AVA_VAN), as this is outside the scope of the CC. Correctness of the implementation of the cryptographic algorithm is considered during the ADV and ATE activities.

B.2.1.4 Monitoring

1888 Information is an abstract view on relation between the properties of entities, i.e. a signal contains information for a system, if the TOE is able to react to

this signal. The TOE resources processes and stores information represented by user data. Therefore:

a) information may flow with the user data between subjects by internal TOE transfer or export from the TOE;

b) information may be generated and passed to other user data;

c) information may be gained through monitoring the operations on data representing the information.

1889 The information represented by user data may be characterised by security attributes like “classification level” having values, for example unclassified, confidential, secret, top secret, to control operations to the data. This information and therefore the security attributes may be changed by operations e.g. FDP_ACC.2 may describe decrease of the level by “sanitarisation” or increase of level by combination of data. This is one aspects of an information flow analysis focused on controlled operations of controlled subjects on controlled objects.

1890 The other aspect is the analysis of ***illicit information flow***. This aspect is more general than the direct access to objects containing user data addressed by the FDP_ACC family. An ***unenforced*** signalling channel carrying information under control of the information flow control policy can also be caused by monitoring of the processing of any object containing or related to this information (e.g. side channels). An ***enforced*** signalling channels may be identified in terms of the subjects manipulating resources and the subject or user that observe such manipulation. Classically, covert channels have been identified as timing or storage channels, according to the resource being modified or modulated. As for other monitoring attacks, the use of the TOE is in accordance with the SFRs.

1891 Covert channels are normally applicable in the case when the TOE has unobservability AND multi-level separation policy requirements. Covert channels may be routinely spotted during vulnerability analysis and design activities, and should therefore be tested. However, generally such monitoring attacks are only identified through specialised analysis techniques commonly referred to as “covert channel analysis”. These techniques have been the subject of much research and there are many papers published on this subject. Guidance for the conduct of covert channel analysis should be sought from the evaluation authority.

1892 *Unenforced* information flow monitoring attacks include passive analysis techniques aiming at disclosure of sensitive internal data of the TOE by operating the TOE in the way that corresponds to the guidance documents.

1893 Side Channel Analysis includes crypt analytical techniques based on physical leakage of the TOE. Physical leakage can occur by timing information, power consumption or power emanation during computation of a TSF. Timing information can be collected also by a remote-attacker (having

network access to the TOE), power based information channels requires that the attacker is in the near-by environment of the TOE.

1894 Eavesdropping techniques include interception of all forms of energy, e.g., electromagnetic or optical emanation of computer displays, not necessarily in the near-field of the TOE.

1895 Monitoring also includes exploits of protocol flaws, e.g., an attack on SSL implementation.

B.2.1.5 Misuse

1896 Misuse may arise from:

a) incomplete guidance documentation;

b) unreasonable guidance;

c) unintended misconfiguration of the TOE;

d) forced exception behaviour of the TOE.

1897 If the guidance documentation is incomplete the user may not know how to operate the TOE in accordance with the SFRs. The evaluator should apply familiarity with the TOE gained from performing other evaluation activities to determine that the guidance is complete. In particular, the evaluator should consider the functional specification. The TSF described in this document should be described in the guidance as required to permit secure administration and use through the TSFI available to human users. In addition, the different modes of operation should be considered to ensure that guidance is provided for all modes of operation.

1898 The evaluator may, as an aid, prepare an informal mapping between the guidance and these documents. Any omissions in this mapping may indicate incompleteness.

1899 The guidance is considered to be unreasonable if it makes demands on the TOE's usage or operational environment that are inconsistent with the ST or unduly onerous to maintain security.

1900 A TOE may use a variety of ways to assist the consumer in effectively using that TOE in accordance with the SFRs and prevent unintentional misconfiguration. A TOE may employ functionality (features) to alert the consumer when the TOE is in a state that is inconsistent with the SFRs, whilst other TOEs may be delivered with enhanced guidance containing suggestions, hints, procedures, etc. on using the existing security features most effectively; for instance, guidance on using the audit feature as an aid for detecting when the SFRs are being compromised; namely insecure.

1901 The evaluator considers the TOE's functionality, its purpose and security objectives for the operational environment to arrive at a conclusion of

whether or not there is reasonable expectation that use of the guidance would permit transition into an insecure state to be detected in a timely manner.

1902 The potential for the TOE to enter into insecure states may be determined using the evaluation deliverables, such as the ST, the functional specification and any other design representations provided as evidence for components included in the assurance package for the TOE (e.g. the TOE/TSF design specification if a component from TOE design (ADV_TDS) is included).

1903 Instances of forced exception behaviour of the TSF could include, but are not limited to, the following:

a) behaviour of the TOE when start-up, close-down or error recovery is activated;

b) behaviour of the TOE under extreme circumstances (sometimes termed overload or asymptotic behaviour), particularly where this could lead to the de-activation or disabling of parts of the TSF;

c) any potential for unintentional misconfiguration or insecure use arising from attacks noted in the section on tampering above.

B.2.2 Identification of Potential Vulnerabilities

1904 Potential vulnerabilities may be identified by the evaluator during different activities. They may become apparent during an evaluation activity or they may be identified as a result of analysis of evidence to search for vulnerabilities.

B.2.2.1 Encountered

1905 The encountered identification of vulnerabilities is where potential vulnerabilities are identified by the evaluator during the conduct of evaluation activities, i.e. the evidence are not being analysed with the express aim of identifying potential vulnerabilities.

1906 The encountered method of identification is dependent on the evaluator's experience and knowledge; which is monitored and controlled by the Certification Authority. It is not reproducible in approach, but will be documented to ensure repeatability of the conclusions from the reported potential vulnerabilities.

1907 There are no formal analysis criteria required for this method. Potential vulnerabilities are identified from the evidence provided as a result of knowledge and experience. However, this method of identification is not constrained to any particular subset of evidence.

1908 Evaluator is assumed to have knowledge of the TOE-type technology and known security flaws as documented in the public domain. The level of knowledge assumed is that which can be gained from a security e-mail list relevant to the TOE type, the regular bulletins (bug, vulnerability and security flaw lists) published by those organisations researching security

issues in products and technologies in widespread use. This knowledge is not expected to extend to specific conference proceedings or detailed theses produced by university research for AVA_VAN.1 or AVA_VAN.2. However, to ensure the knowledge applied is up to date, the evaluator may need to perform a search of public domain material.

1909 For AVA_VAN.3 to AVA_VAN.5 the search of publicly available information is expected to include conference proceeding and theses produced during research activities by universities and other relevant organisations.

1910 Examples of how these may arise (how the evaluator may encounter potential vulnerabilities):

a) while the evaluator is examining some evidence, it sparks a memory of a potential vulnerability identified in a similar product type, that the evaluator believes to also be present in the TOE under evaluation;

b) while examining some evidence, the evaluator spots a flaw in the specification of an interface, that reflects a potential vulnerability.

1911 This may include becoming aware of a potential vulnerability in a TOE through reading about generic vulnerabilities in a particular product type in an IT security publication or on a security e-mail list to which the evaluator is subscribed.

1912 Attack methods can be developed directly from these potential vulnerabilities. Therefore, the encountered potential vulnerabilities are collated at the time of producing penetration tests based on the evaluator's vulnerability analysis. There is no explicit action for the evaluator to encounter potential vulnerabilities. Therefore, the evaluator is directed through an implicit action specified in AVA_VAN.1.2E and AVA_VAN.*.4E.

1913 Current information regarding public domain vulnerabilities and attacks may be provided to the evaluator by, for example, an evaluation authority. This information is to be taken into account by the evaluator when collating encountered vulnerabilities and attack methods when developing penetration tests.

B.2.2.2 Analysis

1914 The following types of analysis are presented in terms of the evaluator actions.

B.2.2.2.1 Unstructured Analysis

1915 The unstructured analysis to be performed by the evaluator (for Evaluation of sub-activity (AVA_VAN.2)) permits the evaluator to consider the generic vulnerabilities (as discussed in B.2.1). The evaluator will also apply their experience and knowledge of flaws in similar technology types.

B.2.2.2.2 Focused

1916 During the conduct of evaluation activities the evaluator may also identify areas of concern. These are specific portions of the TOE evidence that the evaluator has some reservation about, although the evidence meets the requirements for the activity with which the evidence is associated. For example, a particular interface specification looks particularly complex, and therefore may be prone to error either in the development of the TOE or in the operation of the TOE. There is no potential vulnerability apparent at this stage, further investigation is required. This is beyond the bounds of encountered, as further investigation is required.

1917 Difference between potential vulnerability and area of concern:

a) Potential vulnerability - The evaluator knows a method of attack that can be used to exploit the weakness or the evaluator knows of vulnerability information that is relevant to the TOE.

b) Area of concern - The evaluator may be able to discount concern as a potential vulnerability based on information provided elsewhere. While reading interface specification, the evaluator identifies that due to the extreme (unnecessary) complexity of an interface a potential vulnerability may lay within that area, although it is not apparent through this initial examination.

1918 The focused approach to the identification of vulnerabilities is an analysis of the evidence with the aim of identifying any potential vulnerabilities evident through the contained information. It is an unstructured analysis, as the approach is not predetermined. This approach to the identification of potential vulnerabilities can be used during the independent vulnerability analysis required by Evaluation of sub-activity (AVA_VAN.3).

1919 This analysis can be achieved through different approaches, that will lead to commensurate levels of confidence. None of the approaches have a rigid format for the examination of evidence to be performed.

1920 The approach taken is directed by the results of the evaluator's assessment of the evidence to determine it meets the requirements of the AVA/AGD sub-activities. Therefore, the investigation of the evidence for the existence of potential vulnerabilities may be directed by any of the following:

a) areas of concern identified during examination of the evidence during the conduct of evaluation activities;

b) reliance on particular functionality to provide separation, identified during the analysis of the architectural design (as in Evaluation of sub-activity (ADV_ARC.1)), requiring further analysis to determine it cannot be bypassed;

c) representative examination of the evidence to hypothesise potential vulnerabilities in the TOE.

1921 The evaluator will report what actions were taken to identify potential vulnerabilities in the evidence. However, the evaluator may not be able to describe the steps in identifying potential vulnerabilities before the outset of the examination. The approach will evolve as a result of the outcome of evaluation activities.

1922 The areas of concern may arise from examination of any of the evidence provided to satisfy the SARs specified for the TOE evaluation. The information publicly accessible is also considered.

1923 The activities performed by the evaluator can be repeated and the same conclusions, in terms of the level of assurance in the TOE, can be reached although the steps taken to achieve those conclusions may vary. As the evaluator is documenting the form the analysis took, the actual steps taken to achieve those conclusions are also reproducible.

B.2.2.2.3 Methodical

1924 The methodical analysis approach takes the form of a structured examination of the evidence. This method requires the evaluator to specify the structure and form the analysis will take (i.e. the manner in which the analysis is performed is predetermined, unlike the focused identification method). The method is specified in terms of the information that will be considered and how/why it will be considered. This approach to the identification of potential vulnerabilities can be used during the independent vulnerability analysis required by Evaluation of sub-activity (AVA_VAN.4) and Evaluation of sub-activity (AVA_VAN.5).

1925 This analysis of the evidence is deliberate and pre-planned in approach, considering all evidence identified as an input into the analysis.

1926 All evidence provided to satisfy the (ADV) assurance requirements specified in the assurance package are used as input to the potential vulnerability identification activity.

1927 The “methodical” descriptor for this analysis has been used in an attempt to capture the characterisation that this identification of potential vulnerabilities is to take an ordered and planned approach. A “method” or “system” is to be applied in the examination. The evaluator is to describe the method to be used in terms of what evidence will be considered, the information within the evidence that is to be examined, the manner in which this information is to be considered; and the hypothesis that is to be generated.

1928 The following provide some examples that a hypothesis may take:

a) consideration of malformed input for interfaces available to an attacker at the external interfaces;

b) examination of a security mechanism, such as domain separation, hypothesising internal buffer overflows leading to degradation of separation;

c) analysis to identify any objects created in the TOE implementation representation that are then not fully controlled by the TSF, and could be used by an attacker to undermine the SFRs.

1929 For example, the evaluator may identify that interfaces are a potential area of weakness in the TOE and specify an approach to the analysis that "all interface specifications provided in the functional specification and TOE design will be analysed to hypothesise potential vulnerabilities" and go on to explain the methods used in the hypothesis.

1930 This identification method will provide a plan of attack of the TOE, that would be performed by an evaluator completing penetration testing of potential vulnerabilities in the TOE. The rationale for the method of identification would provide the evidence for the coverage and depth of exploitation determination that would be performed on the TOE.

B.3 When attack potential is used

B.3.1 Developer

1931 Attack potential is used by a PP/ST author during the development of the PP/ST, in consideration of the threat environment and the selection of assurance components. This may simply be a determination that the attack potential possessed by the assumed attackers of the TOE is generically characterised as Basic, Enhanced-Basic, Moderate or High. Alternatively, the PP/ST may wish to specify particular levels of individual factors assumed to be possessed by attackers. (e.g. the attackers are assumed to be experts in the TOE technology type, with access to specialised equipment.)

1932 The PP/ST author considers the threat profile developed during a risk assessment (outside the scope of the CC, but used as an input into the development of the PP/ST in terms of the Security Problem Definition or in the case of low assurance STs, the objectives statement). Consideration of this threat profile in terms of one of the approaches discussed in the following sections will permit the specification of the attack potential the TOE is to resist.

B.3.2 Evaluator

1933 Attack potential is especially considered by the evaluator in two distinct ways during the ST evaluation and the vulnerability assessment activities.

1934 Attack potential is used by an evaluator during the conduct of the vulnerability analysis sub-activity to determine whether or not the TOE is resistant to attacks assuming a specific attack potential of an attacker. If the evaluator determines that a potential vulnerability is exploitable in the TOE, they have to confirm that it is exploitable considering all aspects of the intended environment, including the attack potential assumed by an attacker.

1935 Therefore, using the information provided in the threat statement of the Security Target, the evaluator determines the minimum attack potential

required by an attacker to effect an attack, and arrives at some conclusion about the TOE's resistance to attacks. Table 2 demonstrates the 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 2 Vulnerability testing and attack potential

1936 The “beyond high” entry in the residual vulnerabilities column of the above table represents those potential vulnerabilities that would require an attacker to have an attack potential greater than that of “high” in order to exploit the potential vulnerability. A vulnerability classified as residual in this instance reflects the fact that a known weakness exists in the TOE, but in the current operational environment, with the assumed attack potential, the weakness cannot be exploited.

1937 At any level of attack potential a potential vulnerability may be deemed “infeasible” due to a countermeasure in the operational environment that prevents the vulnerability from being exploited.

1938 A vulnerability analysis applies to all TSFI, including ones that access probabilistic or permutational mechanisms. No assumptions are made regarding the correctness of the design and implementation of the TSFI; nor are constraints placed on the attack method or the attacker's interaction with the TOE - if an attack is possible, then it is to be considered during the vulnerability analysis. As shown in Table 2, successful evaluation against a vulnerability assurance component reflects that the TSF is designed and implemented to protect against the required level of threat.

1939 It is not necessary for an evaluator to perform an attack potential calculation for each potential vulnerability. In some cases it is apparent when developing the attack method whether or not the attack potential required to develop and run the attack method is commensurate with that assumed of the attacker in the operational environment. For any vulnerabilities for which an exploitation is determined, the evaluator performs an attack potential calculation to determine that the exploitation is appropriate to the level of attack potential assumed for the attacker.

1940 The approach described below is to be applied whenever it is necessary to calculate attack potential, unless the evaluation authority provides mandatory guidance that an alternative approach is to be applied. The values given in Tables 3 and 4 below are not mathematically proven. Therefore, the values given in these example tables may need to be adjusted according to the

technology type and specific environments. Guidance from the evaluation authority should be sought.

B.4 Calculating attack potential

B.4.1 Application of attack potential

1941 Attack potential is a function of expertise, resources and motivation. There are multiple methods of representing and quantifying these factors. Also, there may be other factors that are applicable for particular TOE types.

B.4.1.1 Treatment of motivation

1942 Motivation is an attack potential factor that can be used to describe several aspects related to the attacker and the assets the attacker desires. Firstly, motivation can imply the likelihood of an attack - one can infer from a threat described as highly motivated that an attack is imminent, or that no attack is anticipated from an un-motivated threat. However, except for the two extreme levels of motivation, it is difficult to derive a probability of an attack occurring from motivation.

1943 Secondly, motivation can imply the value of the asset, monetarily or otherwise, to either the attacker or the asset holder. An asset of very high value is more likely to motivate an attack compared to an asset of little value. However, other than in a very general way, it is difficult to relate asset value to motivation because the value of an asset is subjective - it depends largely upon the value an asset holder places on it.

1944 Thirdly, motivation can imply the expertise and resources with which an attacker is willing to effect an attack. One can infer that a highly motivated attacker is likely to acquire sufficient expertise and resources to defeat the measures protecting an asset. Conversely, one can infer that an attacker with significant expertise and resources is not willing to effect an attack using them if the attacker's motivation is low.

1945 During the course of preparing for and conducting an evaluation, all three aspects of motivation are at some point considered. The first aspect, likelihood of attack, is what may inspire a developer to pursue an evaluation. If the developer believes that the attackers are sufficiently motivated to mount an attack, then an evaluation can provide assurance of the ability of the TOE to thwart the attacker's efforts. Where the operational environment is well defined, for example in a system evaluation, the level of motivation for an attack may be known, and will influence the selection of countermeasures.

1946 Considering the second aspect, an asset holder may believe that the value of the assets (however measured) is sufficient to motivate attack against them. Once an evaluation is deemed necessary, the attacker's motivation is considered to determine the methods of attack that may be attempted, as well as the expertise and resources used in those attacks. Once examined, the developer is able to choose the appropriate assurance level, in particular the

AVA requirement components, commensurate with the attack potential for the threats. During the course of the evaluation, and in particular as a result of completing the vulnerability assessment activity, the evaluator determines whether or not the TOE, operating in its operational environment, is sufficient to thwart attackers with the identified expertise and resources.

1947 It may be possible for a PP author to quantify the motivation of an attacker, as the PP author has greater knowledge of the operational environment in which the TOE (conforming to the requirements of the PP) is to be placed. Therefore, the motivation could form an explicit part of the expression of the attack potential in the PP, along with the necessary methods and measures to quantify the motivation.

B.4.2 Characterising attack potential

1948 This section examines the factors that determine attack potential, and provides some guidelines to help remove some of the subjectivity from this aspect of the evaluation process.

B.4.2.1 Determining the attack potential

1949 The determination of the attack potential for an attack corresponds to the identification of the effort required to create the attack, and to demonstrate that it can be successfully applied to the TOE (including setting up or building any necessary test equipment), thereby exploiting the vulnerability in the TOE. The demonstration that the attack can be successfully applied needs to consider any difficulties in expanding a result shown in the laboratory to create a useful attack. For example, where an experiment reveals some bits or bytes of a confidential data item (such as a key), it is necessary to consider how the remainder of the data item would be obtained (in this example some bits might be measured directly by further experiments, while others might be found by a different technique such as exhaustive search). It may not be necessary to carry out all of the experiments to identify the full attack, provided it is clear that the attack actually proves that access has been gained to a TOE asset, and that the complete attack could realistically be carried out in exploitation according to the AVA_VAN component targeted. In some cases the only way to prove that an attack can realistically be carried out in exploitation according to the AVA_VAN component targeted is to perform completely the attack and this shall be rated. One of the outputs from the identification of a potential vulnerability is assumed to be a script that gives a step-by-step description of how to carry out the attack that can be used in the exploitation of the vulnerability on another instance of the TOE.

1950 In many cases, the evaluators will estimate the parameters for exploitation, rather than carry out the full exploitation. The estimates and their rationale will be documented in the ETR.

B.4.2.2 Factors to be considered

1951 The following factors should be considered during analysis of the attack potential required to exploit a vulnerability:

a) Time taken to identify and exploit (***Elapsed Time***);

b) Specialist technical expertise required (***Specialist Expertise***);

c) Knowledge of the TOE design and operation (***Knowledge of the TOE***);

d) ***Window of opportunity***;

e) ***IT hardware/software or other equipment*** required for exploitation.

1952 In many cases these factors are not independent, but may be substituted for each other in varying degrees. For example, expertise or hardware/software may be a substitute for time. A discussion of these factors follows. (The levels of each factor are discussed in increasing order of magnitude.) When it is the case, the less “expensive” combination shall be considered in exploitation phase.

1953 ***Elapsed time*** is the total amount of time taken by an attacker to identify that a particular potential vulnerability may exist in the TOE, to develop an attack method and to sustain effort required to mount the attack against the TOE. When considering this factor, the worst case scenario should be used to estimate the amount of time required. The identified amount of time is as follows:

a) less than one day;

b) between one day and one week;

c) between one week and two weeks;

d) between two weeks and one month;

e) each additional month up to 6 months leads to an increased value;

f) more than 6 months.

1954 ***Specialist expertise*** refers to the level of generic knowledge of the underlying principles, product type or attack methods (e.g. Internet protocols, Unix operating systems, buffer overflows). The identified levels are as follows:

a) Laymen are unknowledgeable compared to experts or proficient persons, with no particular expertise;

b) Proficient persons are knowledgeable in that they are familiar with the security behaviour of the product or system type;

c) Experts are familiar with the underlying algorithms, protocols, hardware, structures, security behaviour, principles and concepts of security employed, techniques and tools for the definition of new attacks, cryptography, classical attacks for the product type, attack methods, etc. implemented in the product or system type.

d) The level "Multiple Expert" is introduced to allow for a situation, where different fields of expertise are required at an Expert level for distinct steps of an attack.

1955 It may occur that several types of expertise are required. By default, the higher of the different expertises factors is chosen. In very specific cases, the "multiple expert" level could be used but it should be noted that the expertise must concern fields that are strictly different like for example HW manipulation and cryptography.

1956 ***Knowledge of the TOE*** refers to specific expertise in relation to the TOE. This is distinct from generic expertise, but not unrelated to it. Identified levels are as follows:

a) Public information concerning the TOE (e.g. as gained from the Internet);

b) Restricted information concerning the TOE (e.g. knowledge that is controlled within the developer organisation and shared with other organisations under a non-disclosure agreement)

c) Sensitive information about the TOE (e.g. knowledge that is shared between discreet teams within the developer organisation, access to which is constrained only to members of the specified teams);

d) Critical information about the TOE (e.g. knowledge that is known by only a few individuals, access to which is very tightly controlled on a strict need to know basis and individual undertaking).

1957 The knowledge of the TOE may graduate according to design abstraction, although this can only be done on a TOE by TOE basis. Some TOE designs may be public source (or heavily based on public source) and therefore even the design representation would be classified as public or at most restricted, while the implementation representation for other TOEs is very closely controlled as it would give an attacker information that would aid an attack and is therefore considered to be sensitive or even critical.

1958 It may occur that several types of knowledge are required. In such cases, the higher of the different knowledge factors is chosen.

1959 ***Window of opportunity*** (Opportunity) is also an important consideration, and has a relationship to the ***Elapsed Time*** factor. Identification or exploitation of a vulnerability may require considerable amounts of access to a TOE that may increase the likelihood of detection. Some attack methods may require

considerable effort off-line, and only brief access to the TOE to exploit. Access may also need to be continuous, or over a number of sessions.

1960 For some TOEs the ***Window of opportunity*** may equate to the number of samples of the TOE that the attacker can obtain. This is particularly relevant where attempts to penetrate the TOE and undermine the SFRs may result in the destruction of the TOE preventing use of that TOE sample for further testing, e.g. hardware devices. Often in these cases distribution of the TOE is controlled and so the attacker must apply effort to obtain further samples of the TOE.

1961 For the purposes of this discussion:

a) unnecessary/unlimited access means that the attack doesn't need any kind of opportunity to be realised because there is no risk of being detected during access to the TOE and it is no problem to access the number of TOE samples for the attack;

b) easy means that access is required for less than a day and that the number of TOE samples required to perform the attack is less than ten;

c) moderate means that access is required for less than a month and that the number of TOE samples required to perform the attack is less than one hundred;

d) difficult means that access is required for at least a month or that the number of TOE samples required to perform the attack is at least one hundred;

e) none means that the opportunity window is not sufficient to perform the attack (the length for which the asset to be exploited is available or is sensitive is less than the opportunity length needed to perform the attack - for example, if the asset key is changed each week and the attack needs two weeks); another case is, that a sufficient number of TOE samples needed to perform the attack is not accessible to the attacker - for example if the TOE is a hardware and the probability to destroy the TOE during the attack instead of being successful is very high and the attacker has only access to one sample of the TOE.

1962 Consideration of this factor may result in determining that it is not possible to complete the exploit, due to requirements for time availability that are greater than the opportunity time.

1963 ***IT hardware/software or other equipment*** refers to the equipment required to identify or exploit a vulnerability.

a) Standard equipment is readily available to the attacker, either for the identification of a vulnerability or for an attack. This equipment may be a part of the TOE itself (e.g. a debugger in an operating system),

or can be readily obtained (e.g. Internet downloads, protocol analyser or simple attack scripts).

b) Specialised equipment is not readily available to the attacker, but could be acquired without undue effort. This could include purchase of moderate amounts of equipment (e.g. power analysis tools, use of hundreds of PCs linked across the Internet would fall into this category), or development of more extensive attack scripts or programs. If clearly different test benches consisting of specialised equipment are required for distinct steps of an attack this shall be rated as bespoke.

c) Bespoke equipment is not readily available to the public as it may need to be specially produced (e.g. very sophisticated software), or because the equipment is so specialised that its distribution is controlled, possibly even restricted. Alternatively, the equipment may be very expensive.

d) The level "Multiple Bespoke" is introduced to allow for a situation, where different types of bespoke equipment are required for distinct steps of an attack.

1964 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

1965 Table 3 identifies the factors discussed in the previous section and associates numeric values with the total value of each factor.

1966 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 four should be selected according to the evaluator's view of the impact of those design changes. The table is intended as a guide.

1967 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
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

(1) When several proficient persons are required to complete the attack path, the resulting level of expertise still remains “proficient” (which leads to a 3 rating).
(2) Indicates that the attack path is not exploitable due to other measures in the intended operational environment of the TOE.
(3) If clearly different test benches consisting of specialised equipment are required for distinct steps of an attack, this should be rated as bespoke.

Table 3 Calculation of attack potential

1968 To determine the resistance of the TOE to the potential vulnerabilities identified the following steps should be applied:

a) Define the possible attack scenarios {AS1, AS2, ..., ASn} for the TOE in the operational environment.

b) For each attack scenario, perform a theoretical analysis and calculate the relevant attack potential using Table 3.

c) For each attack scenario, if necessary, perform penetration tests in order to confirm or to disprove the theoretical analysis.

d) Divide all attack scenarios {AS1, AS2, ..., ASn} into two groups:

1) the attack scenarios having been successful (i.e. those that have been used to successfully undermine the SFRs), and

2) the attack scenarios that have been demonstrated to be unsuccessful.

e) For each successful attack scenario, apply Table 4 and determine, whether there is a contradiction between the resistance of the TOE and the chosen AVA_VAN assurance component, see the last column of Table 4.

f) Should one contradiction be found, the vulnerability assessment will fail, e.g. the author of the ST chose the component AVA_VAN.5 and an attack scenario with an attack potential of 21 points (high) has broken the security of the TOE. In this case the TOE is resistant to attacker with attack potential 'Moderate', this contradicts to AVA_VAN.5, hence, the vulnerability assessment fails.

1969 The “Values” column of Table 3 indicates the range of attack potential values (calculated using Table 4) of an attack scenario that results in the SFRs being undermined.

Values	Attack potential required to exploit scenario:	TOE resistant to attackers with attack potential of:	Meets assurance components::	Failure of components:
0-9	Basic	No rating	-	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
10-13	Enhanced-Basic	Basic	AVA_VAN.1, AVA_VAN.2	AVA_VAN.3, AVA_VAN.4, AVA_VAN.5
14-19	Moderate	Enhanced-Basic	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3	AVA_VAN.4, AVA_VAN.5
20-24	High	Moderate	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4	AVA_VAN.5
=>25	Beyond High	High	AVA_VAN.1, AVA_VAN.2, AVA_VAN.3, AVA_VAN.4, AVA_VAN.5	-

Table 4 Rating of vulnerabilities and TOE resistance

1970 An approach such as this cannot take account of every circumstance or factor, but should give a better indication of the level of resistance to attack required to achieve the standard ratings. Other factors, such as the reliance on unlikely chance occurrences are not included in the basic model, but can be used by an evaluator as justification for a rating other than those that the basic model might indicate.

1971 It should be noted that whereas a number of vulnerabilities rated individually may indicate high resistance to attack, collectively the combination of vulnerabilities may indicate that overall a lower rating is applicable. The presence of one vulnerability may make another easier to exploit.

1972 If a PP/ST author wants to use the attack potential table for the determination of the level of attack the TOE should withstand (selection of Vulnerability analysis (AVA_VAN) component), he should proceed as follows: For different types of attacker and/or different types of attack the author has in mind, several passes through Table 3 should be made to determine the different values of attack potential understood for each type of attacker. The PP/ST author then considers the highest value obtained in order to determine the claimed level of resistance from Table 4.

B.5 Example calculation for direct attack

1973 Mechanisms subject to direct attack are often vital for system security and developers often strengthen these mechanisms. As an example, a TOE might use a simple pass number authentication mechanism that can be overcome by an attacker who has the opportunity to repeatedly guess another user's pass number. The system can strengthen this mechanism by restricting pass numbers and their use in various ways. During the course of the evaluation an analysis of this direct attack could proceed as follows:

1974 Information gleaned from the ST and design evidence reveals that identification and authentication provides the basis upon which to control access to network resources from widely distributed terminals. Physical access to the terminals is not controlled by any effective means. The duration of access to a terminal is not controlled by any effective means. Authorised users of the system choose their own pass numbers when initially authorised to use the system, and thereafter upon user request. The system places the following restrictions on the pass numbers selected by the user:

a) the pass number must be at least four and no greater than six digits long;

b) consecutive numerical sequences are disallowed (such as 7,6,5,4,3);

c) repeating digits is disallowed (each digit must be unique).

1975 Guidance provided to the users at the time of pass number selection is that pass numbers should be as random as possible and should not be affiliated with the user in some way - a date of birth, for instance.

1976 The pass number space is calculated as follows:

a) Patterns of human usage are important considerations that can influence the approach to searching a password space. Assuming the worst case scenario and the user chooses a number comprising only four digits, the number of pass number permutations assuming that each digit must be unique is:

$$7(8)(9)(10) = 5040$$

b) The number of possible increasing sequences is seven, as is the number of decreasing sequences. The pass number space after disallowing sequences is:

$$5040 - 14 = 5026$$

1977 Based on further information gleaned from the design evidence, the pass number mechanism is designed with a terminal locking feature. Upon the sixth failed authentication attempt the terminal is locked for one hour. The failed authentication count is reset after five minutes so that an attacker can

at best attempt five pass number entries every five minutes, or 60 pass number entries every hour.

1978 On average, an attacker would have to enter 2513 pass numbers, over 2513 minutes, before entering the correct pass number. The average successful attack would, as a result, occur in slightly less than:

$$7(8)(9)(10) = 5040$$

1979 Using either of the approaches to calculate attack potential described above, it is possible that a layman can defeat the mechanism within days (given easy access to the TOE), with the use of standard equipment, and with no knowledge of the TOE, giving a value of 1. Given the resulting sum, 1, the attack potential required to effect a successful attack is not rated, as it falls below that considered to be Basic.