Common Methodology for Information Technology Security Evaluation

---

Evaluation Methodology

March 2004

Version 2.4
Revision 256

ASE/APE Trial Use version

CCIMB-2004-03-004

Foreword

This document, version 2.4 of the Common Methodology for Information Technology Security Evaluation (CEM), is issued for use by the international IT security evaluation community. The CEM is a companion document to the Common Criteria for Information Technology Security Evaluation (CC) and is the result of extensive international cooperation. Practical experience acquired through use in evaluations, and requests for interpretations received, will be used to further develop the CEM

This Legal NOTICE has been placed in the CEM by request:

The seven governmental organisations (collectively called the "Common Criteria Project Sponsoring Organisations") listed just below, as the joint holders of the copyright in the Common Methodology for Information Technology Security Evaluations, version 2.4 (called "CEM 2.4"), hereby grant non-exclusive license to ISO/IEC to use CEM 2.4 in the continued development/maintenance of the ISO/IEC 18045 international standard. However, the Common Criteria Project Sponsoring Organisations retain the right to use, copy, distribute, translate or modify CEM 2.4 as they see fit.

Canada: Communications Security Establishment
France: Service Central de la Sécurité des Systèmes d'Information
Germany: Bundesamt für Sicherheit in der Informationstechnik
Netherlands: Netherlands National Communications Security Agency
United Kingdom: Communications-Electronics Security Group
United States: National Institute of Standards and Technology
United States: National Security Agency

Table of Contents

Table of contents

Table of contents

List of figures

1 Introduction

1.1 Scope

1 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.

2 The scope of this version is limited to evaluations of Protection Profiles and TOEs for EAL1 and EAL4, as defined in the CC. It does not provide guidance for EALs 2, 3 , and 5 through 7, nor for evaluations using other assurance packages. The CEM is based on CC version 2.4

3 The target audience for the 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.

4 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 B.

5 This revision of the CEM consists of only one part. It supersedes all older versions and parts of the CEM.

1.2 Organisation

6 This CEM is divided into the following chapters:

7 Chapter 1, Introduction describes the objectives, organisation, document conventions and terminology, and evaluator verdicts.

8 Chapter 2, Evaluation process and related tasks describes the tasks that are relevant for all evaluation activities. These are the tasks used to manage the inputs and prepare the outputs.

9 Chapter 3, Protection Profile evaluation describes the methodology for the evaluation of Protection Profiles, based on the APE class of CC Part 3.

10 Chapters 4 and 5, describe the evaluation methodology for the Evaluation Assurance Levels EAL1 and EAL4 defined in CC Part 3.

11 Chapter 6, Flaw remediation sub-activities, describes the methodology for the evaluation of the Flaw remediation (ALC_FLR) family of CC Part 3.

12 Annex A, Glossary, defines vocabulary and references used in the CEM and presents abbreviations and acronyms.

13 Annex B, General evaluation guidance, provides guidance common to several activities described in Chapters 4 through 5.

1.3 Document Conventions

1.3.1 Terminology

14 The glossary, presented in Annex A of this part, includes only those terms used in a specialised way within this document. The majority of terms are used according to their accepted definitions.

15 The term *activity* is used to describe the application of an assurance class of the CC Part 3.

16 The term *sub-activity* is used to describe 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.

17 The term *action* is related to an 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.

18 The term *work unit* is 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 *4:ALC_TAT.1-2*. In this symbol, the first digit (*4*) indicates the EAL; 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.

19 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. For example, because an additional work unit labeled 4:ADV_FSP.2-7 was added at EAL4, the subsequent sequential numbering of FSP work units is offset by one. Thus work unit 3:ADV_FSP.1-8 is now mirrored by work unit 4:ADV_FSP.2-9; each express the same requirement though their numbering no longer directly correspond.

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

1.3.2 Verb usage

21 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.

22 Guidance text accompanying work units and sub-tasks gives further explanation on how to apply the CC words in an evaluation. The described method is normative, meaning that the verb usage is in accordance with ISO definitions for these verbs; that is: the auxiliary verb *should* is used when the described method is strongly preferred and the auxiliary verb *may* is used where the described method(s) is allowed but no preference is indicated. (The auxiliary verb *shall* is used only for the text of work units.)

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

1.3.3 General evaluation guidance

24 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 B. 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.

1.3.4 Relationship between CC and CEM structures

25 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.

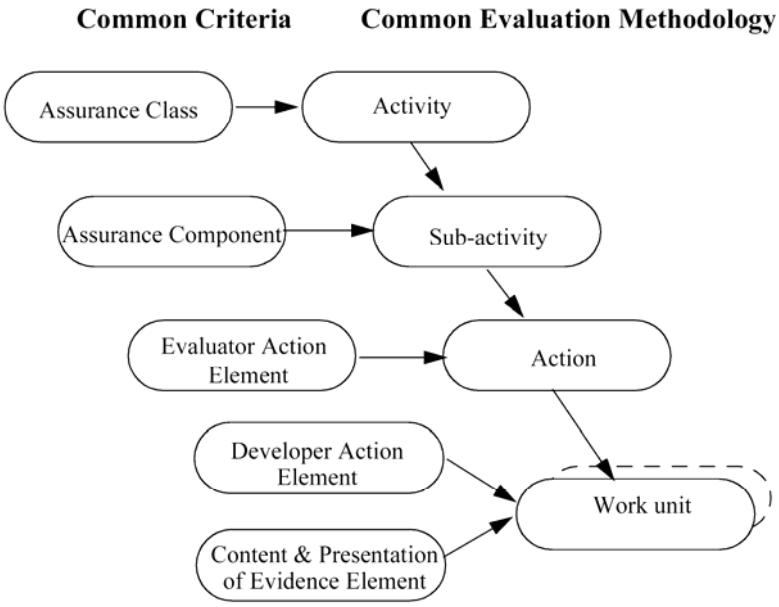

Figure 1 - Mapping of the CC and CEM structures

2 Evaluation process and related tasks

2.1 Introduction

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

27 Each evaluation, whether of a PP or TOE (including ST), follows the same process, and has three evaluator tasks in common: the input task, the output task and the evaluation sub-activities.

28 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).

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

30 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.

2.2 Evaluation process overview

2.2.1 Objectives

31 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.

2.2.2 Responsibilities of the roles

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

33 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.

34 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.

35 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.

36 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.

2.2.3 Relationship of roles

37 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.

38 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.

2.2.4 General evaluation model

39 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.

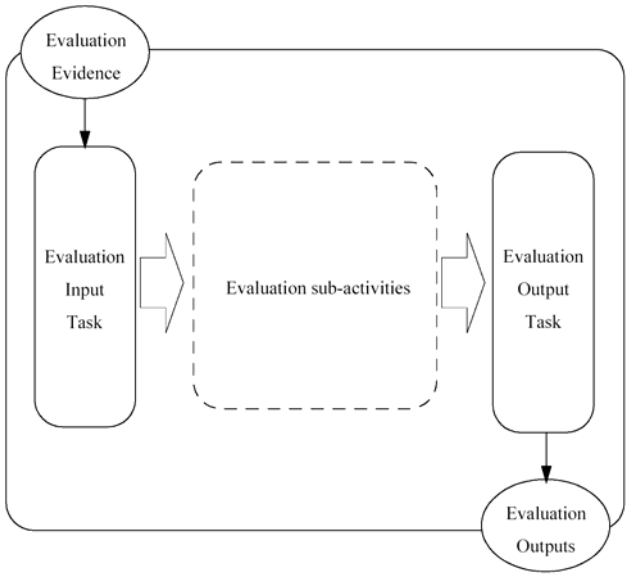

Figure 2 - Generic evaluation model

40 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.

2.2.5 Evaluator verdicts

41 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, Section 5.3.

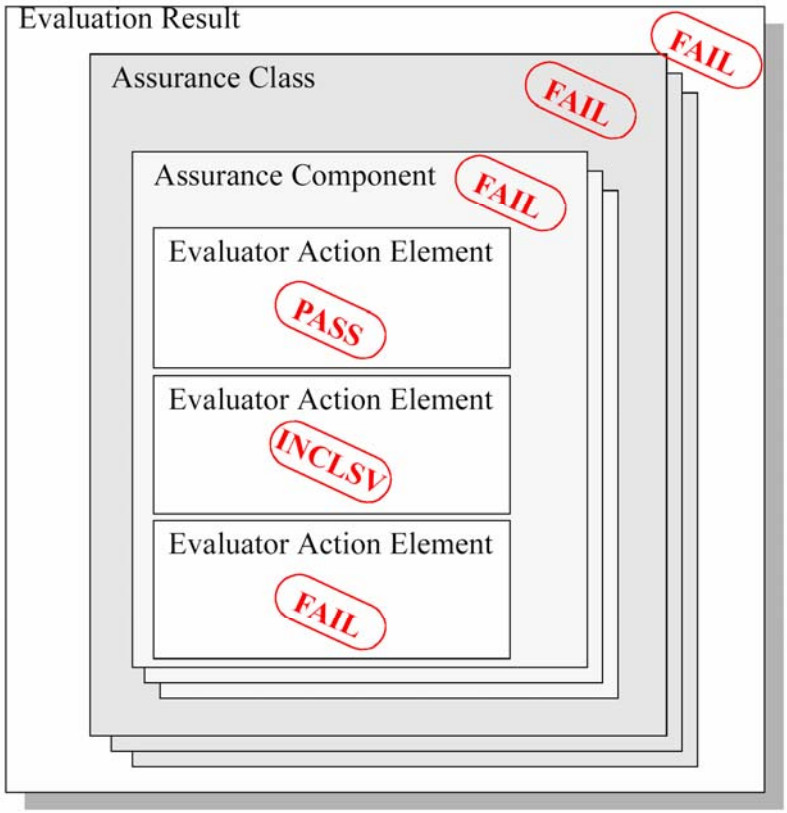

Figure 3 - Example of the verdict assignment rule

42 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 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* of *fail* verdict is assigned.

43 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*.

2.3 Evaluation input task

2.3.1 Objectives

44 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.

2.3.2 Application notes

45 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.

46 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 a high-level design is required, then the High-level design (ADV_HLD) 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 (ACM_CAP) and Delivery (ADO_DEL)) will also apply to the entire TOE (including any part produced by another developer).

47 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.

48 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.

49 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.

50 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).

2.3.3 Management of evaluation evidence sub-task

2.3.3.1 Configuration control

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

52 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.

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

2.3.3.2 Disposal

54 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.

2.3.3.3 Confidentiality

55 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.

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

2.4 Evaluation sub-activities

57 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.

58 Each of the Chapters 4 through 5 is organised similarly based on the evaluation work required for an evaluation. Chapter Protection Profile evaluation addresses the work necessary for reaching an evaluation result on a PP.

2.5 Evaluation output task

2.5.1 Objectives

59 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.

60 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.

61 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.

2.5.2 Management of evaluation outputs

62 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.

2.5.3 Application notes

63 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. The information resulting from evaluator work to assist in re-evaluation or re-use has not yet been determined by the CEMEB under their current work program. 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.

2.5.4 Write OR sub-task

64 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.

65 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.

66 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.

67 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).

2.5.5 Write ETR sub-task

2.5.5.1 Objectives

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

69 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 clauses) be reported in the ETR.

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

71 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.

2.5.5.2 ETR for a PP Evaluation

72 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.

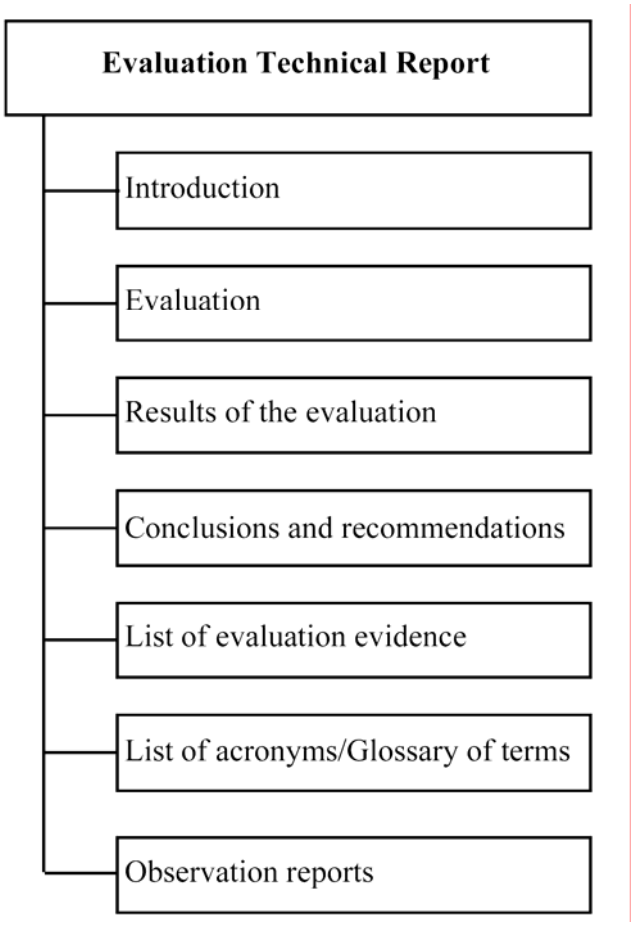

Figure 4 - ETR information content for a PP evaluation

2.5.5.2.1 Introduction

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

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

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

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

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

78 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.

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

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

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

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

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

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

2.5.5.2.2 Evaluation

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

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

87 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.

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

2.5.5.2.3 Results of the evaluation

89 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.

90 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.

2.5.5.2.4 Conclusions and recommendations

91 The evaluator ***shall report*** the conclusions of the evaluation, in particular the overall verdict as defined in CC Part 1 Chapter 5, and determined by application of the verdict assignment described in 2.2.5.

92 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.

2.5.5.2.5 List of evaluation evidence

93 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).

2.5.5.2.6 List of acronyms/Glossary of terms

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

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

2.5.5.2.7 Observation reports

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

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

2.5.5.3 ETR for a TOE Evaluation

98 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.

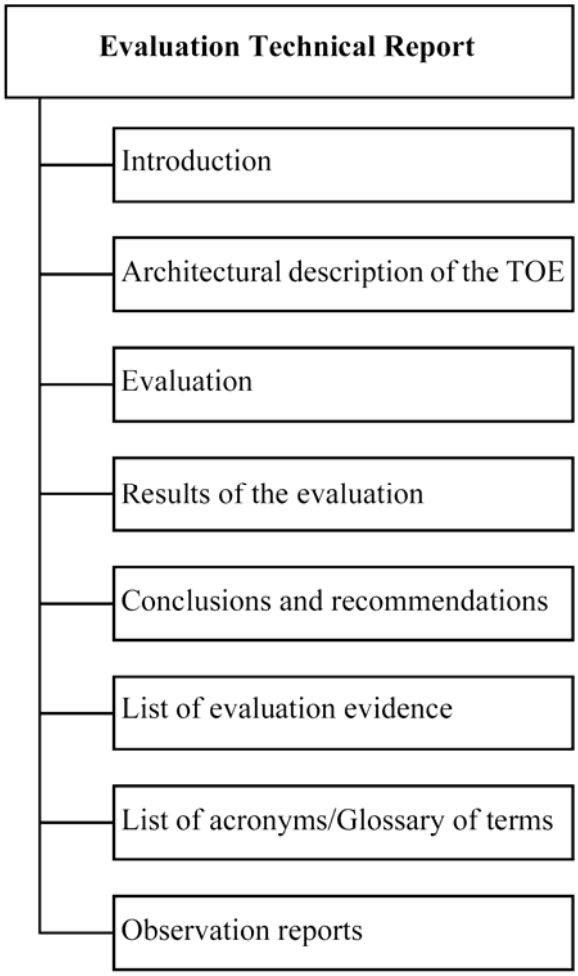

Figure 5 - ETR information content for a TOE evaluation

2.5.5.3.1 Introduction

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

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

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

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

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

104 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.

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

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

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

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

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

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

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

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

2.5.5.3.2 Architectural description of the TOE

113 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 “Development - High-level design (ADV_HLD)”, where applicable.

114 The intent of this section is to characterise the degree of architectural separation of the major components. If there is no High-level design (ADV_HLD) requirement in the ST, this is not applicable and is considered to be satisfied.

2.5.5.3.3 Evaluation

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

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

117 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.

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

2.5.5.3.4 Results of the evaluation

119 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.

120 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.

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

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

2.5.5.3.5 Conclusions and recommendations

123 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 5, and determined by application of the verdict assignment described in 2.2.5.

124 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.

2.5.5.3.6 List of evaluation evidence

125 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).

2.5.5.3.7 List of acronyms/Glossary of terms

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

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

2.5.5.3.8 Observation reports

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

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

3 Protection Profile evaluation

3.1 Introduction

130 This clause 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 clause is based on the requirements on the PP as specified in CC Part 3 class APE.

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

3.2 Objectives

132 The PP is the description of a TOE type. As such it is expected to identify the security requirements that enforce the defined OSPs and counter the defined threats under the defined assumptions.

133 Evaluating a PP is required to demonstrate that the PP is sound and internally consistent, and, if the PP is based on one or more PPs or packages, that the PP is a correct instantiation of these PPs or packages. These properties are necessary for the PP to be suitable for use as the basis for an ST.

3.3 PP evaluation relationships

134 The activities to conduct a complete PP evaluation cover the following:

a) evaluation input task (Chapter 2);

b) PP evaluation activity, comprising the following sub-activities:

1) evaluation of the PP introduction (Section 3.4.3);

2) evaluation of the conformance claims (Section 3.4.1);

3) evaluation of the security problem definition (Section 3.4.7);

4) evaluation of the security objectives (Section 3.4.4);

5) evaluation of the extended security requirements (Section 3.4.2);

6) evaluation of the stated security requirements (Section 3.4.5);

7) evaluation of the derived security requirements (Section 3.4.6);

8) evaluation output task (Chapter 2).

135 The evaluation input and evaluation output tasks are described in Section 2. The evaluation activities are derived from the APE assurance requirements contained in CC Part 3.

136 The sub-activities comprising a PP evaluation are described in this section. Although the sub-activities can, in general, be started more or less coincidentally, some dependencies between sub-activities have to be considered by the evaluator.

137 Some of the information required for the PP may be included by reference. For example if compliance to a PP is claimed, some information in the PP such as the threats may be included by reference only. All material that is referred to in such a way is considered to be part of the PP and should conform to the APE criteria.

3.4 Protection Profile evaluation activity

3.4.1 Evaluation of Conformance claims (APE_CCL.1)

3.4.1.1 Objectives

138 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.

3.4.1.2 Input

139 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.

3.4.1.3 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.

140 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 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.

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

142 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.

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

144 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.

145 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).

146 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.

147 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).

148 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 the conformance claim states a claim of either package-name conformant or package-name augmented.

149 If the package conformance claim contains package-name conformant, the evaluator determines that the PP contains no security requirements in addition to those included in the package.

150 If the package conformance claim contains package-name augmented, the evaluator determines that the PP includes at least one security requirement in addition to those included in the package.

APE_CCL.1.7C ***The conformance claims 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.

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

152 The relation between the types could be simple: a firewall ST claiming conformance to a firewall PP, or more complex: a smartcard ST claiming conformance to a number of PPs at the same time: a PP for the integrated circuit, a PP for the smartcard OS, and two PPs for two applications on the smartcard.

APE_CCL.1.8C ***The conformance claims 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.

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

154 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP for which conformance is being claimed.

In this instance the statement of SPD must be stated in exactly the same wording as that used in the PP for which conformance is being claimed. The PP under evaluation may repeat any threats, OSPs and/or assumptions or it may include them by reference to the PP they come from.

155 Where strict or demonstrable conformance is required by the PP, the conformance claim rationale should provide a tracing between the statement of SDP in the PP under evaluation and that in the PP for which conformance is being claimed. This tracing should be sufficient for the evaluator to determine that all threats, assumptions and OSPs detailed in the PP are represented in the PP under evaluation.

156 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the PP author is allowed to add threats, OSPs and/or assumptions to those drawn from those in the PPs for which conformance is being claimed.

APE_CCL.1.9C ***The conformance claims rationale shall demonstrate that the statement of objectives is consistent with the statement of 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 PP, with the statement of security objectives in the PPs.

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

158 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP for which conformance is being claimed. In this instance the security objectives must be stated in exactly the same wording as that used in the PP for which conformance is being claimed. The PP under evaluation may repeat any security objective, or it may include it by reference to the PP it comes from.

159 Where strict or demonstrable conformance is required by the PP for which conformance is being claimed, the conformance claim rationale should provide a tracing between the statement of security objectives in the PP under evaluation and that in the PP for which conformance is being claimed. This tracing should be sufficient for the evaluator to determine that all security objectives detailed in the PP are represented in the PP under evaluation.

160 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add objectives to those drawn from those in the PP for which conformance is being claimed.

APE_CCL.1.10C ***The conformance claims 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.

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

162 The PP under evaluation may repeat any security requirements or it may include them by reference to the PP(s) they come from. If, however, the security requirements in the PP for which conformance is being claimed include uncompleted operations, or the author of the PP under evaluation has applied the refinement operation on any PP security requirements, then these security requirements must be fully present in the PP under evaluation.

163 For exact conformance, the conformance rationale will be trivial, as the statement of security requirements in the PP under evaluation must include the same requirements as in the PPs for which conformance is being claimed, with no additions, deletions or substitutions.

164 For strict conformance, the conformance rationale will be trivial again; demonstrating that the statement of requirements in the PP is a non-strict super set of those in the PP. That is, that all requirements in the PP for which conformance is being claimed have been included in the PP under evaluation, possibly with some additional requirements.

165 For demonstrable conformance, the evaluator determines that the justification for the security requirements in the PP for which conformance is being claimed demonstrates that each requirement is represented by one or more security requirements in the PP under evaluation.

166 The evaluator is also reminded that if strict or demonstrable conformance with PPs is required, the author of the PP under evaluation is allowed to add security requirements to those drawn from those PPs for which conformance is being claimed.

APE_CCL.1.11C ***The conformance claims rationale shall demonstrate that all operations of the security requirements that were taken from a PP are completed consistently with the respective PP.***

APE_CCL.1-13 The evaluator ***shall examine*** the conformance claim rationale to determine that that the completion of the security requirements is consistent with those in the PP for which conformance is being claimed.

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

168 The PP, for which conformance is being claimed, may already have partially completed operations in a requirement, or set other limits on the completion of those operations. If this is the case the evaluator determines that the corresponding requirement in the PP under evaluation is completed consistent with these partial completions and/or limits.

169 An example of an inconsistent completion is a PP that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The PP under evaluation that claims conformance to this PP, copies the requirement and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

170 Note that, if the PP for which conformance is being claimed in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the PP under evaluation with any number other than 5 would be an inconsistent completion.

APE_CCL.1.12C ***The conformance claims rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the security requirements package for which conformance is being claimed.***

APE_CCL.1-14 The evaluator ***shall examine*** the PP to determine that it is consistent with all security requirements in the packages for which conformance is being claimed.

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

172 The PP may repeat any security requirements or it may include them by reference to the package(s) they come from. If, however, the package security requirements include uncompleted operations, or the PP author has applied the refinement operation on any package security requirements, then these security requirements must be fully present in the PP.

173 The evaluator is also reminded that if the conformance claim is package-name augmented the PP author is permitted to add security requirements to those drawn from that package.

APE_CCL.1.13C ***The conformance claims rationale shall demonstrate that all operations of the security requirements in the PP that were taken from a package are completed consistently with the respective security requirement package.***

APE_CCL.1-15 The evaluator ***shall examine*** the PP to determine that all security requirements in the PP that were taken from a security requirements package is completed consistently with that security requirements package.

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

175 If the security requirements package has already partially completed operations in a requirement, or has set other limits on the completion of those operations, the evaluator determines that the corresponding requirement in the PP is completed consistent with these partial completions and/or limits.

176 An example of an inconsistent completion is a package that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "The TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "The TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The PP that claims conformance to this package, copies the requirement in the PP and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

177 Note that, if the security requirements package in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the PP with any number other than 5 would be an inconsistent completion.

APE_CCL.1.14C ***The conformance statement shall describe the conformance required of any PPs/STs as exact-PP, strict-PP or demonstrable-PP -conformance for the PP.***

APE_CCL.1-16 The evaluator ***shall check*** that the PP conformance statement states a claim of exact-PP, strict-PP, demonstrable-PP -conformance

3.4.2 Evaluation of Extended components definition (APE_ECD.1)

3.4.2.1 Objectives

178 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 could not have been clearly expressed using existing CC Part 2 or CC Part 3 components.

3.4.2.2 Input

179 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.2.3 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 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.

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

181 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.

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

183 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

184 If the extended component is a member of an existing CC Part 2 or 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.

185 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.

186 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

187 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.

188 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.

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

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

APE_ECD.1-5 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended functional component identifies all applicable audit information of that component.

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

192 The evaluator confirms that no applicable security relevant events that are candidates for audit have been overlooked by the PP author.

APE_ECD.1-6 The evaluator ***shall examine*** the extended security requirement components definition to determine that each definition of an extended functional component identifies all applicable security management information of that component.

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

194 The evaluator confirms that no applicable security management functions for this component 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.1-7 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.

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

196 The evaluator determines that the extended functional component is consistent with CC Part 2 Section 2.1.3 Component structure.

197 If the extended functional component uses operations, the evaluator determines that the extended functional component is consistent with CC Part 1 Section 4.4.1.3 Component.

198 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 2.2.1 Component changes highlighting.

APE_ECD.1-8 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.

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

200 The evaluator determines that all new functional families are defined consistent with CC Part 2 Section 2.1.2 Family structure.

APE_ECD.1-9 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.

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

202 The evaluator determines that all new functional classes are defined consistent with CC Part 2 Section 2.1.1 Class structure

APE_ECD.1-10 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.

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

204 The evaluator determines that the extended assurance component definition is consistent with CC Part 3 Section 2.1.3 Assurance component structure.

205 If the extended assurance component uses operations, the evaluator determines that the extended assurance component is consistent with CC Part 1 Section 4.4.1.3 Component.

206 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 2.1.3 Assurance component structure.

APE_ECD.1-11 The evaluator ***shall examine*** the extended components definition to determine that for each defined extended assurance component, applicable methodology has been provided.

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

208 The evaluator determines that for each evaluator action element of each extended SAR one or more work units is provided and that succesfully performing all work units for a given evaluator action element will demonstrate that the element has been achieved.

APE_ECD.1-12 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.

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

210 The evaluator determines that all new assurance families are defined consistent with CC Part 3 Section 2.1.2 Assurance family structure.

APE_ECD.1-13 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.

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

212 The evaluator determines that all new assurance classes are defined consistent with CC Part 3 Section 2.1.1 Class structure.

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

APE_ECD.1-14 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 compliance or noncompliance can be demonstrated.

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

214 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.

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

216 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 requirements are to be used as a model for determining what constitutes compliance with this requirement.

3.4.2.4 Action APE_ECD.1.2E

APE_ECD.1-15 The evaluator ***shall examine*** the extended components definition to determine that each extended component can not be clearly expressed using existing components.

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

218 The evaluator determines that each extended component cannot be clearly expressed using existing components. The evaluator should take components from CC Part 2 and 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.

219 The evaluator is reminded that the role of this work unit is to preclude unnecessary duplication of components, that is, components that can be clearly expressed 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 with existing components.

3.4.3 Evaluation of PP introduction (APE_INT.1)

3.4.3.1 Objectives

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

3.4.3.2 Input

221 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.3.3 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.

222 The evaluator determines that the PP reference identifies the PP itself, so that it can 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.

223 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.

224 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.

225 The evaluator determines whether 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.

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

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

3.4.4 Evaluation of Security objectives (APE_OBJ.1)

3.4.4.1 Objectives

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

3.4.4.2 Input

229 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.4.3 Action APE_OBJ.1.1E

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

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

230 The evaluator checks that the security objectives for the TOE are identified, and that they are clearly separated from the security objectives for the development environment and the security objectives for the operational environment.

APE_OBJ.1.2C ***The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs met by that security objective.***

APE_OBJ.1-2 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 organisational policies met by the objectives.

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

232 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.1.3C ***The statement of security objectives shall describe the security objectives for the development environment.***

APE_OBJ.1-3 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the development environment

233 The evaluator checks that the security objectives for the development environment are identified, and that they are also clearly separated from the security objectives for the TOE and the security objectives for the operational environment.

APE_OBJ.1.4C ***The security objectives rationale shall trace each security objective for the development environment back to threats countered by that security objective and OSPs met by that security objective.***

APE_OBJ.1-4 The evaluator ***shall check*** that the security objectives rationale traces the security objectives for the development environment back to threats countered by that security objective and OSPs met by that security objective.

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

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

APE_OBJ.1.5C ***The statement of security objectives shall describe the security objectives for the operational environment***

APE_OBJ.1-5 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the operational environment.

236 The evaluator checks that the security objectives for the operational environment are identified, and that they are also clearly separated from the

security objectives for the TOE and the security objectives for the development environment.

APE_OBJ.1.6C ***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.1-6 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.

237 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.

238 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.1.7C ***The security objectives rationale shall demonstrate that the security objectives counter all threats.***

APE_OBJ.1-7 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.

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

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

241 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 threats are sufficiently mitigated.

242 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 could be as minimal as “Security Objective X directly counters threat Y”.

243 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.1.8C ***The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.***

APE_OBJ.1-8 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.

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

245 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 implemented.

246 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 implementation of the OSP.

247 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 could be as minimal as "Security Objective X directly enforces OSP Y".

APE_OBJ.1.9C ***The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.***

APE_OBJ.1-9 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.

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

249 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 is consistent with the assumption.

250 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 achieving consistency with the assumption.

251 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 could be as minimal as "Security Objective for the Operational Environment X directly upholds Assumption Y".

3.4.4.4 Action APE_OBJ.1.2E

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

252 The evaluator should compare the security objectives with each other to determine whether they contradict each other, or whether there may be conditions in which they contradict each other.

253 Examples of such contradictions are:

– "a user's identity shall never be released" and "actions of a user shall be logged with that user's identity".

– "the network connection in the operational environment shall be 100% available" and "The network connection in the operational environment shall fail in a secure manner by shutting down its services gracefully".

– "it hall not be possible for type X users to access type Y data", "type X users shall be able to export type Y data out of the TOE" may contradict unless the type Y data is protected in another way.

3.4.5 Evaluation of Security requirements (APE_REQ.1)

3.4.5.1 Objectives

254 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and canonically formulated and whether they are internally consistent.

3.4.5.2 Input

255 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.5.3 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.

256 The evaluator determines that all SFRs are 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 compliant with;

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

e) by reference to an individual component in the PP.

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

258 If an SFR is reproduced in the PP, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 2.

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

259 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 PP;

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

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

e) by reproduction in the PP.

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

261 If an SAR is reproduced in the PP, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 3.

APE_REQ.1.2C ***The statement of security requirements shall identify all operations on the security requirements.***

APE_REQ.1-3 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

262 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 can be achieved by typographical distinctions, or by explicit identification in the surrounding text, or by any other distinctive means.

APE_REQ.1.3C ***All operations shall be performed correctly.***

APE_REQ.1-4 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

263 An assignment operation is only allowed where specifically permitted in a component.

264 The evaluator compares each assignment with the component from which it is derived to determine that the assignment has been left uncompleted, has been completed, or has been transformed into a selection.

265 If the assignment has been left uncompleted, the evaluator determines that the uncompleted assignment has been correctly copied from the component.

266 If the assignment has been completed with a value, the evaluator determines that the type of the value is consistent with the type required by the assignment. An assignment may only be completed with "None" if this is specifically allowed by the component.

267 If the assignment has been transformed into a selection, the evaluator determines that each value in the selection is an allowable value of the assignment. "None" may only be a choice in the selection if "None" is specifically allowed by the component as a completion of the assignment.

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

268 The evaluator determines that each iteration of a requirement is different from each other iteration of that requirement (at least one element is different), or that the requirement applies to a different part of the TOE.

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

269 A selection operation is only allowed where specifically permitted in a component.

270 The evaluator compares each selection with the component from which it is derived to determine that the selection has been left uncompleted, has been completed, or has been restricted.

271 If the selection has been left uncompleted, the evaluator determines that the uncompleted selection has been correctly copied from the component.

272 If the selection has been completed, the evaluator determines that selected item or items are one or more of the items indicated within the selection portion of the component. The evaluator also determines that where a selection explicitly states "choose one of" only one item is selected.

273 If the selection has been restricted, the evaluator determines that the remaining selectable items are a subset of the selectable items in the component.

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

274 The evaluator determines for each refinement that the component is refined in such manner that a TOE meeting the refined requirement also meets the unrefined requirement. If the refined requirement exceeds this boundary it is considered to be an extended requirement.

275 A special case of refinement is an editorial refinement, where a small change is made in a requirement, i.e. rephrasing a sentence due to adherence to proper English grammar. This change is not allowed to modify the meaning of the requirement in any way. The evaluator is reminded that editorial refinements have to be clearly identified.

276 Another special case of refinement is where multiple iterations of the same requirement are used, each with different refinements, where some of the refined iterations do not meet the full scope of the original requirement. This is acceptable, provided that all iterations of the refined requirement taken collectively, meet the entire scope of the original requirement.

277 In addition, a refinement should be related to the original requirement. Refining an audit requirement with an extra element on prevention of electromagnetic radiation is normally not allowed. This refinement should be added to another requirement, or if no applicable requirement to refine can be found, be formulated as an extended requirement.

3.4.5.4 Action APE_REQ.1.2E

APE_REQ.1-8 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

279 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 “all objects”, “all subjects” etc., that these requirements do not conflict.

280 Some possible conflicts are:

a) FRU_RSA.2 Minimum and maximum quotas specifying a minimum number of resources available to a user and FTA_MCS.1 Basic limitation on multiple concurrent sessions specifying a maximum number of sessions available for a user. If the resources are somehow linked to sessions, these requirements may conflict;

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

c) FPR_ANO.1 Anonymity specifying anonymity, FAU_GEN.1 Audit data generation specifying that subject identity is to be logged, and FAU_SAR.1 Audit review specifying who can read the audit records. If people from whom the activities of users should be hidden, can read the audit logs of these activities, these requirements may conflict;

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

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

3.4.6 Evaluation of Security requirements (APE_REQ.2)

3.4.6.1 Objectives

281 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and canonically formulated, whether they are internally consistent, and whether they meet the security objectives of the TOE and the security objectives for the development environment.

3.4.6.2 Input

282 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.6.3 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.

283 The evaluator determines that all SFRs are 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 compliant with;

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

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

e) by reproduction in the PP.

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

285 If an SFR is reproduced in the PP, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 2.

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

286 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 PP;

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

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

e) by reproduction in the PP.

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

288 If an SAR is reproduced in the PP, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 3.

APE_REQ.2.2C ***The statement of security requirements shall identify all operations on the security requirements.***

APE_REQ.2-3 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

289 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 can be achieved by typographical distinctions, or by explicit identification in the surrounding text, or by any other distinctive means.

APE_REQ.2.3C ***All operations shall be performed correctly.***

APE_REQ.2-4 The evaluator ***shall examine*** the statement of security requirements to determine that all assignment operations are performed correctly.

290 An assignment operation is only allowed where specifically permitted in a component.

291 The evaluator compares each assignment with the component from which it is derived to determine that the assignment has been left uncompleted, has been completed, or has been transformed into a selection.

292 If the assignment has been left uncompleted, the evaluator determines that the uncompleted assignment has been correctly copied from the component.

293 If the assignment has been completed with a value, the evaluator determines that the type of the value is consistent with the type required by the assignment. An assignment may only be completed with "None" if this is specifically allowed by the component.

294 If the assignment has been transformed into a selection, the evaluator determines that each value in the selection is an allowable value of the assignment. "None" may only be a choice in the selection if "None" is specifically allowed by the component as a completion of the assignment.

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

295 The evaluator determines that each iteration of a requirement is different from each other iteration of that requirement (at least one element is different), or that the requirement applies to a different part of the TOE.

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

296 A selection operation is only allowed where specifically permitted in a component.

297 The evaluator compares each selection with the component from which it is derived to determine that the selection has been left uncompleted, has been completed, or has been restricted.

298 If the selection has been left uncompleted, the evaluator determines that the uncompleted selection has been correctly copied from the component.

299 If the selection has been completed, the evaluator determines that selected item or items are one or more of the items indicated within the selection portion of the component. The evaluator also determines that where a selection explicitly states "choose one of" only one item is selected.

300 If the selection has been restricted, the evaluator determines that the remaining selectable items are a subset of the selectable items in the component.

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

301 The evaluator determines for each refinement that the component is refined in such manner that a TOE meeting the refined requirement also meets the unrefined requirement. If the refined requirement exceeds this boundary it is considered to be an extended requirement.

302 A special case of refinement is an editorial refinement, where a small change is made in a requirement, i.e. rephrasing a sentence due to adherence to proper English grammar. This change is not allowed to modify the meaning of the requirement in any way. The evaluator is reminded that editorial refinements have to be clearly identified.

303 Another special case of refinement is where multiple iterations of the same requirement are used, each with different refinements, where some of the refined iterations do not meet the full scope of the original requirement. This is acceptable, provided that all iterations of the refined requirement taken collectively, meet the entire scope of the original requirement.

304 In addition, a refinement should be related to the original requirement. Refining an audit requirement with an extra element on prevention of electromagnetic radiation is normally not allowed. This refinement should be added to another requirement, or if no applicable requirement to refine can be found, be formulated as an extended requirement.

APE_REQ.2.4C ***Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.***

APE_REQ.2-8 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.

305 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.

306 A justification that a dependency is not met can 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.2.5C ***The security requirements rationale shall trace each SFR back to the security objectives for the TOE.***

APE_REQ.2-9 The evaluator ***shall check*** that the security requirements rationale traces each SFR back to the security objectives for the TOE.

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

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

APE_REQ.2.6C ***The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.***

APE_REQ.2-10 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.

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

310 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.

311 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.

312 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.

313 The evaluator takes into account that the TOE should usually have some form of protection for itself, otherwise it will not be able to uphold its security objectives. After all, if the TSF itself can be corrupted, it will not perform its duties for long in a hostile environment.

314 While examining the justification, the evaluator takes into account that SFRs can be bypassed, tampered with, deactivated, or attacked without being detected, and that this may lead to the security objectives for the TOE not being achieved. In particular, the evaluator closely examines cases where:

a) Reference mediation (FPT_RVM) is not included, as this indicates possible bypass;

b) Domain separation (FPT_SEP) is not included, as this indicates possible logical tampering;

c) TSF physical protection (FPT_PHP) is not included, as this indicates possible physical tampering;

d) FAU: Security audit components are not included, as this indicates that attacks can be performed without being detected;

e) FMT: Security management components have been included, as this provides a possibility to modify the behaviour of other SFRs,

315 and these cases are not or not sufficiently addressed by the security objectives for the operational environment.

APE_REQ.2.7C ***The security requirements rationale shall trace each SAR back to the security objectives for the development environment.***

APE_REQ.2-11 The evaluator ***shall check*** that the security requirements rationale traces each SAR back to the security objectives for the development environment.

316 The evaluator determines that each SAR is traced back to at least one security objective for the development environment.

317 Failure to trace implies that either the security requirements rationale is incomplete, the security objectives for the development environment are incomplete, or that the SAR has no useful purpose.

APE_REQ.2.8C ***The security requirements rationale shall demonstrate that the SARs meet all security objectives for the development environment.***

APE_REQ.2-12 The evaluator ***shall examine*** the security requirements rationale to determine that for each security objective for the development environment it justifies that the SARs are suitable to meet that security objective for the development environment.

318 If no SARs trace back to the security objective for the development environment, this work unit fails.

319 The evaluator determines that the justification for a security objective for the development environment demonstrates that the SARs are sufficient: if all SARs that trace back to the objective are satisfied, the security objective for the development environment is achieved.

320 If the SARs that trace back to a security objective for the development environment 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.

321 The evaluator also determines that each SAR that traces back to a security objective for the development environment is necessary, when the SAR is satisfied, it actually contributes to achieving the security objective.

322 Note that the tracings from SARs to security objectives for the development environment provided in the security requirements rationale may be a part of the justification, but do not constitute a justification by themselves.

3.4.6.4 Action APE_REQ.2.2E

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

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

324 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 "all objects", "all subjects" etc., that these requirements do not conflict.

325 Some possible conflicts are:

a) FRU_RSA.2 Minimum and maximum quotas specifying a minimum number of resources available to a user and FTA_MCS.1 Basic limitation on multiple concurrent sessions specifying a maximum number of sessions available for a user. If the resources are somehow linked to sessions, these requirements may conflict;

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

c) FPR_ANO.1 Anonymity specifying anonymity, FAU_GEN.1 Audit data generation specifying that subject identity is to be logged, and FAU_SAR.1 Audit review specifying who can read the audit records. If people from whom the activities of users should be hidden, can read the audit logs of these activities, these requirements may conflict;

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

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

3.4.7 Evaluation of Security problem definition (APE_SPD.1)

3.4.7.1 Objectives

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

3.4.7.2 Input

327 The evaluation evidence for this sub-activity is:

a) the PP.

3.4.7.3 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.

328 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.

329 The evaluator determines that the security problem definition describes the threats that must be countered by the TOE, its development environment, its operational environment or combinations of these three.

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.

330 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.

331 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.

332 If all security objectives are derived from assumptions and 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.

333 The evaluator determines that OSP statements are made in terms of rules, practices or guidelines that must be followed by the TOE, its development environment, its operational environment or combinations of these three.

334 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.

335 If the threats and/or OSPs already sufficiently address the physical, personnel, and connectivity aspects of the TOE, this work unit is not applicable and is therefore considered to be satisfied.

336 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.

4 EAL1 evaluation

4.1 Introduction

337 EAL1 provides a basic level of assurance. The TOE is analysed using a functional specification and guidance documentation to understand its security behaviour. Independent testing of the TSF against a subset of the SFRs is performed.

4.2 Objectives

338 The objective of this chapter is to define the minimal evaluation effort for achieving an EAL1 evaluation and to provide guidance on ways and means of accomplishing the evaluation.

4.3 EAL1 evaluation relationships

339 An EAL1 evaluation covers the following:

a) evaluation input task (Chapter 2);

b) EAL1 evaluation activities comprising the following:

1) evaluation of the ST (Section 4.4);

2) evaluation of the configuration management (Section 4.5);

3) evaluation of the delivery and operation documents (Section 4.6);

4) evaluation of the development documents (Section 4.7);

5) evaluation of the guidance documents (Section 4.8);

6) testing (Section 4.9);

c) evaluation output task (Chapter 2).

340 The evaluation activities are derived from the EAL1 assurance requirements contained in CC Part 3.

341 The ST evaluation is started prior to any TOE evaluation sub-activities since the ST provides the basis and context to perform these sub-activities.

342 The sub-activities comprising an EAL1 evaluation are described in this chapter. Although the sub-activities can, in general, be started more or less coincidentally, some dependencies between sub-activities have to be considered by the evaluator.

4.4 Security Target evaluation activity

343 This section 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.

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

4.4.1 Application notes

4.4.1.1 ST evaluation relationships

345 The activities to conduct a complete ST evaluation cover the following:

a) evaluation input task (Section 2);

b) ST evaluation activity, comprising the following sub-activities:

1) evaluation of the ST introduction (Section 4.4.4);

2) evaluation of the conformance claims (Section 4.4.2);

3) evaluation of the extended security requirements 4.4.3);

4) evaluation of the security requirements (Section 4.4.5);

5) evaluation of the TOE summary specification (Section 4.4.6).

c) evaluation output task (Section 2).

346 The evaluation input and evaluation output tasks are described in Section 2. The evaluation activities are derived from the ASE assurance requirements contained in CC Part 3.

347 The sub-activities comprising an ST evaluation are described in this clause. Although the sub-activities can, in general, be started more or less coincidentally, some dependencies between sub-activities have to be considered by the evaluator.

348 Some of the information required for the ST may be included by reference. For example if compliance to a PP is claimed, some information in the PP such as the threats may be included by reference only. All material that is referred to in such a way is considered to be part of the ST and should conform to the ASE criteria.

4.4.1.2 Re-using the evaluation results of certified PPs

349 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.

350 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.

351 An example would be where the PP contained 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.

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

4.4.2 Evaluation of Conformance claims (ASE_CCL.1)

4.4.2.1 Objectives

353 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.

4.4.2.2 Input

354 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.

4.4.2.3 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.***

1: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.

355 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.

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.***

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

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.***

1: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.***

1: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.

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

357 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.

1: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.

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

359 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.***

1: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.

360 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).

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

1: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.

362 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).

363 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.***

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

364 If the package conformance claim contains package-name conformant, the evaluator determines that the ST contains no security requirements in addition to those included in the package.

365 If the package conformance claim contains package-name augmented, the evaluator determines that the ST includes at least one security requirement in addition to those included in the package.

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

1: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.

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

367 The relation between the types could be simple: a firewall ST claiming conformance to a firewall PP, or more complex: a smartcard ST claiming conformance to a number of PPs at the same time: a PP for the integrated circuit, a PP for the smartcard OS, and two PPs for two applications on the smartcard.

ASE_CCL.1.8C ***The conformance claims 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.***

1: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.

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

369 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP. In this instance the statement of SPD must be stated in exactly the same wording as that used in the PP. The ST may repeat any threats, OSPs and/or assumptions or it may include them by reference to the PP they come from.

370 Where strict or demonstrable conformance is required by the PP, the conformance claim rationale should provide a tracing between the statement of SDP in the ST and that in the PP. This tracing should be sufficient for the evaluator to determine that all threats, assumptions and OSPs detailed in the PP are represented in the ST.

371 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add threats, OSPs and/or assumptions to those drawn from those in the PPs.

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

1: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.

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

373 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP. In this instance the security objectives must be stated in exactly the same wording as that used in the PP. The ST may repeat any security objective, or it may include it by reference to the PP it comes from.

374 Where strict or demonstrable conformance is required by the PP, the conformance claim rationale should provide a tracing between the statement of security objectives in the ST and that in the PP. This tracing should be

sufficient for the evaluator to determine that all security objectives detailed in the PP are represented in the ST.

375 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add objectives to those drawn from those in the PPs.

ASE_CCL.1.10C ***The conformance claims 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.***

1: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.

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

377 The ST may repeat any security requirements or it may include them by reference to the PP(s) they come from. If, however, the PP security requirements include uncompleted operations, or the ST author has applied the refinement operation on any PP security requirements, then these security requirements must be fully present in the ST.

378 For exact conformance, the conformance rationale will be trivial, as the statement of security requirements in the ST must include the same requirements as in the PPs, with no additions, deletions or substitutions.

379 For strict conformance, the conformance rationale will be trivial again; demonstrating that the statement of requirements in the ST is a non-strict super set of those in the PP. That is, that all requirements in the PP have been included in the ST, possibly with some additional requirements.

380 For demonstrable conformance, the evaluator determines that the justification for the security requirements in the PP demonstrates that each requirement is represented by one or more security requirements in the ST.

381 The evaluator is also reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add security requirements to those drawn from those PPs.

ASE_CCL.1.11C ***The conformance claims rationale shall demonstrate that all operations of the security requirements that were taken from a PP are completed consistently with the respective PP.***

1:ASE_CCL.1-13 The evaluator ***shall examine*** the conformance claim rationale to determine that that the completion of the security requirements in the ST are consistent, in the manner specified in the PP, with those in the PP.

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

383 The PP may already have partially completed operations in a requirement, or set other limits on the completion of those operations. If this is the case, the evaluator determines that the corresponding requirement in the ST is completed consistent with these partial completions and/or limits.

384 An example of an inconsistent completion is a PP that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "The TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "The TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The ST that claims conformance to this PP, copies the requirement in the ST and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

385 Note that, if the PP in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the ST with any number other than 5 would be an inconsistent completion.

ASE_CCL.1.12C ***The conformance claims rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the security requirement package for which conformance is being claimed.***

1:ASE_CCL.1-14 The evaluator ***shall examine*** the ST to determine that it is consistent with all security requirements in the packages for which conformance is being claimed.

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

387 The ST may repeat any security requirements or it may include them by reference to the package(s) they come from. If, however, the package security requirements include uncompleted operations, or the ST author has applied the refinement operation on any package security requirements, then these security requirements must be fully present in the ST.

388 The evaluator is also reminded that if the conformance claim is package-name augmented the ST author is permitted to add security requirements to those drawn from that package.

ASE_CCL.1.13C ***The conformance claims rationale shall demonstrate that all operations of the security requirements in the ST that were taken from a package are completed consistently with the respective security requirement package.***

1:ASE_CCL.1-15 The evaluator ***shall examine*** the ST to determine that all security requirements in the ST that were taken from a security requirements package or PP are completed consistently with that security requirements package or PP.

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

390 If the security requirements package has already partially completed operations in a requirement, or has set other limits on the completion of those operations, the evaluator determines that the corresponding requirement in the ST is completed consistent with these partial completions and/or limits.

391 An example of an inconsistent completion is a package that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "The TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "The TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The ST that claims conformance to this package, copies the requirement in the ST and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

392 Note that, if the security requirements package in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the ST with any number other than 5 would be an inconsistent completion.

4.4.3 Evaluation of Extended components definition (ASE_ECD.1)

4.4.3.1 Objectives

393 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 could not have been clearly expressed using existing CC Part 2 or CC Part 3 components.

4.4.3.2 Input

394 The evaluation evidence for this sub-activity is:

a) the ST.

4.4.3.3 Action ASE_ECD.1.1E

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

1: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 Part 3.

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

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

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

396 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.***

1: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.

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

398 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

399 If the extended component is a member of an existing CC Part 2 or 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.

400 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.

401 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

402 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.

403 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.

1: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.

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

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

1:ASE_ECD.1-5 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended functional component identifies all applicable audit information of that component.

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

407 The evaluator confirms that no applicable security relevant events that are candidates for audit have been overlooked by the ST author.

408 For guidance on audit information of a component, see CC Part 2, Section 2.1.2.5

1:ASE_ECD.1-6 The evaluator ***shall examine*** the extended security requirement components definition to determine that each definition of an extended functional component identifies all applicable security management information of that component.

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

410 The evaluator confirms that no applicable security management functions for this component have been overlooked by the ST author.

411 For guidance on security management information of a component, see CC Part 2, Section 2.1.2.4

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

1:ASE_ECD.1-7 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.

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

413 The evaluator determines that the extended functional component is consistent with CC Part 2 Section 2.1.3 Component structure.

414 If the extended functional component uses operations, the evaluator determines that the extended functional component is consistent with CC Part 1 Section 4.4.1.3 Component.

415 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 2.2.1 Component changes highlighting.

1:ASE_ECD.1-8 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.

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

417 The evaluator determines that all new functional families are defined consistent with CC Part 2 Section 2.1.2 Family structure.

1:ASE_ECD.1-9 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.

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

419 The evaluator determines that all new functional classes are defined consistent with CC Part 2 Section 2.1.1 Class structure.

1:ASE_ECD.1-10 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.

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

421 The evaluator determines that the extended assurance component definition is consistent with CC Part 3 Section 2.1.3 Assurance component structure.

422 If the extended assurance component uses operations, the evaluator determines that the extended assurance component is consistent with CC Part 1 Section 4.4.1.3 Component.

423 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 2.1.3 Assurance component structure.

1:ASE_ECD.1-11 The evaluator ***shall examine*** the extended components definition to determine that for each defined extended assurance component, applicable methodology has been provided.

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

425 The evaluator determines that for each evaluator action element of each extended SAR one or more work units is provided and that succesfully performing all work units for a given evaluator action element will demonstrate that the element has been achieved.

1:ASE_ECD.1-12 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.

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

427 The evaluator determines that all new assurance families are defined consistent with CC Part 3 Section 2.1.2 Assurance family structure.

1:ASE_ECD.1-13 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.

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

429 The evaluator determines that all new assurance classes are defined consistent with CC Part 3 Section 2.1.1 Class structure.

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

1:ASE_ECD.1-14 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 compliance or noncompliance can be demonstrated.

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

431 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.

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

433 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 requirements are to be used as a model for determining what constitutes compliance with this requirement.

4.4.3.4 Action ASE_ECD.1.2E

1:ASE_ECD.1-15 The evaluator ***shall examine*** the extended components definition to determine that each extended component can not be clearly expressed using existing components.

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

435 The evaluator determines that each extended component cannot be clearly expressed using existing components. The evaluator should take components from CC Part 2 and 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.

436 The evaluator is reminded that the role of this work unit is to preclude unnecessary duplication of components, that is, components that can be clearly expressed 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 with existing components.

4.4.4 Evaluation of ST introduction (ASE_INT.1)

4.4.4.1 Objectives

437 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.

4.4.4.2 Input

438 The evaluation evidence for this sub-activity is:

a) the ST.

4.4.4.3 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.***

1: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.***

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

439 The evaluator determines that the ST reference identifies the ST itself, so that it can 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.

440 In evaluations where a CM system is provided, the evaluator could 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.***

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

441 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.

442 This work unit is limited to the TOE reference in the ST, checking whether the TOE is actually labelled with this reference, and whether these references are consistent, is covered by the ACM_CAP CM capabilities family.

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

443 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 are 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.***

1: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.

444 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.

445 The evaluator determines whether 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.***

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

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

446 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.

447 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 can not 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.***

1: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.

448 While some TOEs can 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 is therefore considered to be satisfied.

449 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 should be 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 and boundaries of the TOE.***

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

450 The evaluator determines that the TOE description discusses the hardware, firmware and software components and/or modules that constitute the TOE at a level of detail that is sufficient to give the reader a general understanding of those components and/or modules.

451 The evaluator also determines that the TOE description lists all guidance that is part of the TOE.

452 The evaluator also determines that the TOE description describes exactly where the boundary lies between the TOE hardware/software/firmware and any non-TOE hardware/software/firmware.

453 The evaluator also determines that the TOE description describes exactly where the boundary between the TOE guidance and any non-TOE guidance lies.

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

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

454 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.

455 The evaluator also determines that the TOE description describes exactly where the boundary lies between functionality provided by the TOE, and functionality provided by any non-TOE hardware/software/firmware.

4.4.4.4 Action ASE_INT.1.2E

1: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.

4.4.5 Evaluation of Security requirements (ASE_REQ.1)

4.4.5.1 Objectives

456 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and canonically formulated and whether they are internally consistent.

4.4.5.2 Input

457 The evaluation evidence for this sub-activity is:

a) the ST.

4.4.5.3 Action ASE_REQ.1.1E

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

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

458 The evaluator determines that all SFRs are 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 compliant with;

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

e) by reproduction in the ST.

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

460 If an SFR is reproduced in the ST, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 2.

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

461 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 compliant with;

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

e) by reproduction in the ST.

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

463 If an SAR is reproduced in the ST, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 3.

ASE_REQ.1.2C ***The statement of security requirements shall identify all operations on the security requirements.***

1:ASE_REQ.1-3 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

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

ASE_REQ.1.3C ***All assignment and selection operations shall be completed.***

1:ASE_REQ.1-4 The evaluator ***shall examine*** the statement of security requirements to determine that each assignment and each selection operation is completed.

465 The evaluator determines that there are no choices left in the assignments and selections of all SFRs and all SARs.

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

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

466 An assignment operation is only allowed where specifically permitted in a component.

467 The evaluator compares each assignment with the component from which it is derived to determine that the values of the parameters or variables chosen comply with the indicated type required by the assignment. An assignment may only be completed with “None” if this is specifically allowed.

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

468 The evaluator determines that each iteration of a requirement is different from each other iteration of that requirement (at least one element is different), or that the requirement applies to a different part of the TOE.

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

469 A selection operation is only allowed where specifically permitted in a component.

470 The evaluator compares each selection with the component from which it is derived to determine that the selected item or items are one or more of the items indicated within the selection portion of the component. The evaluator also determines that where a selection explicitly states “choose one of”, only one item is selected.

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

471 The evaluator determines for each refinement that the component is refined in such manner that a TOE meeting the refined requirement also meets the unrefined requirement. If the refined requirement exceeds this boundary it is considered to be an extended requirement.

472 A special case of refinement is an editorial refinement, where a small change is made in a requirement, i.e. rephrasing a sentence due to adherence to proper English grammar. This change is not allowed to modify the meaning of the requirement in any way. The evaluator is reminded that editorial refinements have to be clearly identified.

473 Another special case of refinement is where multiple iterations of the same requirement are used, each with different refinements, where some of the refined iterations do not meet the full scope of the original requirement. This is acceptable, provided that all iterations of the refined requirement taken collectively, meet the entire scope of the original requirement.

474 In addition, a refinement should be related to the original requirement. Refining an audit requirement with an extra element on prevention of

electromagnetic radiation is normally not allowed. This refinement should be added to another requirement, or if no applicable requirement to refine can be found, be formulated as an extended requirement.

4.4.5.4 Action ASE_REQ.1.2E

1:ASE_REQ.1-9 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

476 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 “all objects”, “all subjects” etc., that these requirements do not conflict.

477 Some possible conflicts are:

a) FRU_RSA.2 Minimum and maximum quotas specifying a minimum number of resources available to a user and FTA_MCS.1 Basic limitation on multiple concurrent sessions specifying a maximum number of sessions available for a user. If the resources are somehow linked to sessions, these requirements may conflict;

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

c) FPR_ANO.1 Anonymity, FAU_GEN.1 Audit data generation specifying that subject identity is to be logged, and FAU_SAR.1 Audit review specifying who can read the audit records. If people from whom the activities of users should be hidden, can read the audit logs of these activities, these requirements may conflict;

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

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

4.4.6 Evaluation of TOE summary specification (ASE_TSS.1)

4.4.6.1 Objectives

478 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.

4.4.6.2 Input

479 The evaluation evidence for this sub-activity is:

a) the ST.

4.4.6.3 Action ASE_TSS.1.1E

ASE_TSS.1.1C ***The TOE summary specification shall describe how the TOE meets each SFR.***

1:ASE_TSS.1-1 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE meets each SFR.

480 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.

481 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.

4.4.6.4 Action ASE_TSS.1.2E

1: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.

482 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.

4.5 Configuration management activity

483 The purpose of the configuration management activity is to assist the consumer in identifying the evaluated TOE.

4.5.1 Evaluation of CM capabilities (ACM_CAP.1)

4.5.1.1 Objectives

484 The objectives of this sub-activity are to determine whether the developer has clearly identified the TOE.

4.5.1.2 Input

485 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing.

4.5.1.3 Action ACM_CAP.1.1E

ACM_CAP.1.1C ***The reference for the TOE shall be unique to each version of the TOE.***

1:ACM_CAP.1-1 The evaluator ***shall check*** that the version of the TOE provided for evaluation is uniquely referenced.

486 For this assurance component there is no requirement for the developer to use a CM system, beyond unique referencing. As a result the evaluator is able to verify the uniqueness of a TOE version only by checking that other versions of the TOE available for purchase do not possess the same reference. In evaluations where a CM system was provided in excess of the CC requirements, the evaluator could validate the uniqueness of the reference by checking the configuration list. Evidence that the version provided for evaluation is uniquely referenced may be incomplete if only one version is examined during the evaluation, and the evaluator should look for a referencing system that is capable of supporting unique references (e.g. use of numbers, letters or dates). However, the absence of any reference will normally lead to a fail verdict against this requirement unless the evaluator is confident that the TOE can be uniquely identified.

487 The evaluator should seek to examine more than one version of the TOE (e.g. during rework following discovery of a vulnerability), to check that the two versions are referenced differently.

ACM_CAP.1.2C ***The TOE shall be labelled with its reference.***

1:ACM_CAP.1-2 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

488 The evaluator should ensure that the TOE contains a unique reference such that it is possible to distinguish different versions of the TOE. 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).

489 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.

1:ACM_CAP.1-3 The evaluator ***shall check*** that the TOE references used are consistent.

490 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.

491 The evaluator also verifies that the TOE reference is consistent with the ST.

4.6 Delivery and operation activity

492 The purpose of the delivery and operation activity is to judge the adequacy of the documentation of the procedures used to ensure that the TOE is installed, generated, and started in the same way the developer intended it to be.

4.6.1 Evaluation of Installation, generation and start-up (ADO_IGS.1)

4.6.1.1 Objectives

493 The objective of this sub-activity is to determine whether the procedures and steps for the secure installation, generation, and start-up of the TOE have been documented and result in a secure configuration.

4.6.1.2 Application notes

494 The installation, generation, and start-up procedures refer to all installation, generation, and start-up procedures, regardless of whether they are performed at the user's site or at the development site that are necessary to progress the TOE to the secure configuration as described in the ST.

4.6.1.3 Input

495 The evaluation evidence for this sub-activity is:

a) the administrator guidance;

b) the secure installation, generation, and start-up procedures;

c) the TOE suitable for testing.

4.6.1.4 Action ADO_IGS.1.1E

ADO_IGS.1.1C ***The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.***

1:ADO_IGS.1-1 The evaluator ***shall check*** that the procedures necessary for the secure installation, generation and start-up of the TOE have been provided.

496 If it is not anticipated that the installation, generation, and start-up procedures will or can be reapplied (e.g. because the TOE may already be

delivered in an operational state) this work unit (or the effected parts of it) is not applicable, and is therefore considered to be satisfied.

4.6.1.5 Action ADO_IGS.1.2E

1:ADO_IGS.1-2 The evaluator ***shall examine*** the provided installation, generation, and start-up procedures to determine that they describe the steps necessary for secure installation, generation, and start-up of the TOE.

497 If it is not anticipated that the installation, generation, and start-up procedures will or can be reapplied (e.g. because the TOE may already be delivered in an operational state) this work unit (or the effected parts of it) is not applicable, and is therefore considered to be satisfied.

498 The installation, generation, and start-up procedures may provide detailed information about the following:

a) changing the installation specific security characteristics of entities under the control of the TSF;

b) handling exceptions and problems;

c) minimum system requirements for secure installation if applicable.

499 In order to confirm that the installation, generation, and start-up procedures result in a secure configuration, the evaluator may follow the developer's procedures and may perform the activities that customers are usually expected to perform to install, generate, and start-up the TOE (if applicable to the TOE), using the supplied guidance documentation only. This work unit might be performed in conjunction with the ATE_IND.1-2 work unit.

4.7 Development activity

500 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. This understanding is achieved through examination of a functional specification (which describes the external interfaces of the TOE) and a representation correspondence (which maps the functional specification to the SFRs) in order to ensure consistency).

4.7.1 Application notes

501 The CC requirements for design documentation are levelled by formality. The CC considers a document's degree of formality (that is, whether it is informal, semiformal or formal) 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.

502 An informal functional specification comprises a description of the purpose and method of use of the externally-visible 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 functions. If there is also audit functionality that detects and records 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.

503 Informality of the demonstration of correspondence need not be in a prose form; a simple two-dimensional mapping may be sufficient. For example, a matrix with modules listed along one axis and subsystems listed along the other, with the cells identifying the correspondence of the two, would serve to provide an adequate informal correspondence between the high-level design and the low-level design

4.7.2 Evaluation of Functional specification (ADV_FSP.1)

4.7.2.1 Objectives

504 The objective of this sub-activity is to determine whether the developer has provided an adequate description of the TSF and whether this shows that all SFRs have been sufficiently addressed.

4.7.2.2 Input

505 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance.

4.7.2.3 Action ADV_FSP.1.1E

ADV_FSP.1.1C ***The functional specification shall describe the TSF and its external interfaces using an informal style.***

1:ADV_FSP.1-1 The evaluator ***shall examine*** the functional specification to determine that it contains all necessary informal explanatory text.

506 If the entire functional specification is informal, this work unit is not applicable and is therefore considered to be satisfied.

507 Supporting narrative descriptions are necessary for those portions of the functional specification that are difficult to understand only from the

semiformal or formal description (for example, to make clear the meaning of any formal notation).

ADV_FSP.1.2C ***The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.***

1:ADV_FSP.1-2 The evaluator ***shall examine*** the functional specification to determine that it identifies all of the external TSF interfaces.

508 The term external refers to that which is visible to the user. External interfaces to the TOE are either direct interfaces to the TSF or interfaces to non-TSF portions of the TOE. However, these non-TSF interfaces might have eventual access to the TSF. These external interfaces that directly or indirectly access the TSF collectively make up the TOE security function interface (TSFI). Figure **6** shows a TOE with TSF (cross-hatched) portions and non-TSF (empty) portions. This TOE has three external interfaces: interface c is a direct interface to the TSF; interface b is an indirect interface to the TSF; and interface a is an interface to non-TSF portions of the TOE. Therefore, interfaces b and c make up the TFSI.

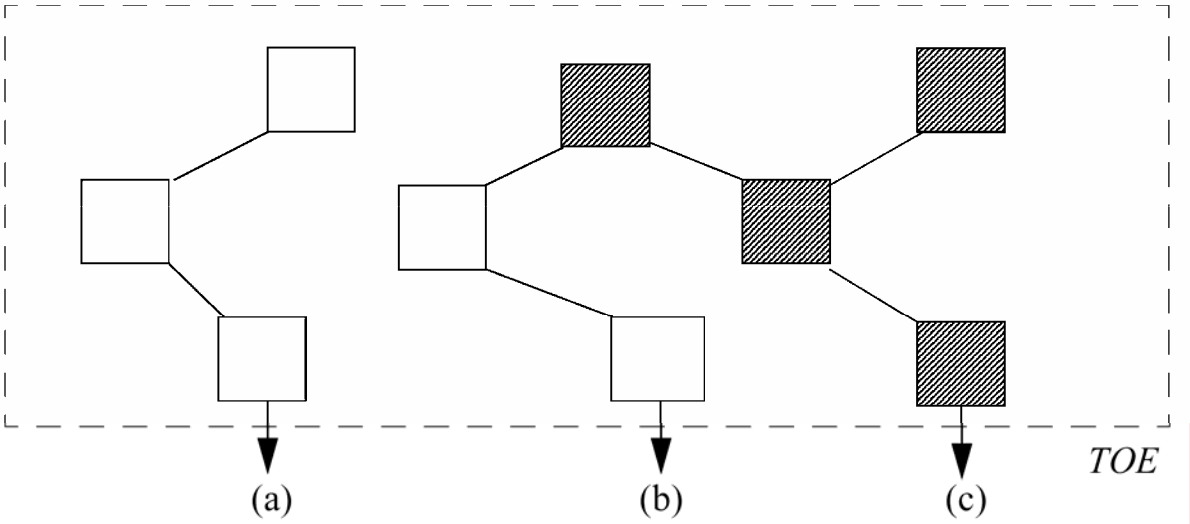

Figure 6 - TSF Interfaces

509 It should be noted that all security functions reflected in the functional requirements of CC Part 2 (or in extended components thereof) will have some sort of externally-visible manifestation. While not all of these are necessarily interfaces from which the security function can be tested, they are all externally-visible to some extent and must therefore be included in the functional specification.

1:ADV_FSP.1-3 The evaluator ***shall examine*** the functional specification to determine that it identifies all of the external TSF interfaces.

510 For a TOE that has no threat of malicious users (i.e. TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation

(FPT_SEP) are rightfully excluded from its ST), the only interfaces that are described in the functional specification (and expanded upon in the other TSF representation descriptions) are those to and from the TSF. The absence of TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) presumes there is no concern for any sort of bypassing of the security features; therefore, there is no concern with any possible impact that other interfaces might have on the TSF.

511 On the other hand, if the TOE has a threat of malicious users or bypass (i.e. TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) are included in its ST), all external interfaces are described in the functional specification, but only to the extent that the effect of each is made clear: interfaces to the security functions (i.e. interfaces b and c in Figure **6**) are completely described, while other interfaces are described only to the extent that it is clear that the TSF is inaccessible through the interface (i.e. that the interface is of type a, rather than b in Figure **6**). The inclusion of TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) implies a concern that all interfaces might have some effect upon the TSF. Because each external interface is a potential TSF interface, the functional specification must contain a description of each interface in sufficient detail so that an evaluator can determine whether the interface is security relevant.

512 Some architectures lend themselves to readily provide this interface description in sufficient detail for groups of external interfaces. For example, a kernel architecture is such that all calls to the operating system are handled by kernel programs; any calls that might violate the TSP must be called by a program with the privilege to do so. All programs that execute with privilege must be included in the functional specification. Any program external to the kernel that executes without privilege is incapable of violating the TSP (i.e. such programs are interfaces of type *a*, rather than *b* in Figure **6**) and may, therefore, be excluded from the functional specification. It is worth noting that, while the evaluator's understanding of the interface description can be expedited in cases where there is a kernel architecture, such an architecture is not necessary.

1:ADV_FSP.1-4 The evaluator ***shall examine*** the presentation of the TSFI to determine that it adequately and correctly describes the behaviour of the TOE at each external interface describing effects, exceptions and error messages.

513 In order to assess the adequacy and correctness of an interface's presentation, the evaluator uses the functional specification and the user and administrator guidance to assess the following factors:

a) All security relevant user input parameters (or a characterisation of those parameters) should be identified. For completeness, parameters outside of direct user control should be identified if they are usable by administrators.

b) All security relevant behaviour described in the reviewed guidance should be reflected in the description of semantics in the functional

specification. This should include an identification of the behaviour in terms of events and the effect of each event. For example, if an operating system provides a rich file system interface, where it provides a different error code for each reason why a file is not opened upon request (e.g. access denied, no such file, file is in use by another user, user is not authorised to open the file after 5pm, etc.), the functional specification should explain that a file is either opened upon request, or else that an error code is returned. (While the functional specification may enumerate all these different reasons for errors, it need not provide such detail.) The description of the semantics should include how the security requirements apply to the interface (e.g. whether the use of the interface is an auditable event and, if so, the information that can be recorded).

c) All interfaces are described for all possible modes of operation. If the TSF provides the notion of privilege, the description of the interface should explain how the interface behaves in the presence or absence of privilege.

d) The information contained in the descriptions of the security relevant parameters and syntax of the interface should be consistent across all documentation.

514 Verification of the above is done by reviewing the SFRs and the functional specification, as well as the user and administrator guidance provided by the developer. For example, if the TOE were an operating system and its underlying hardware, the evaluator would look for discussions of user-accessible programs, descriptions of protocols used to direct the activities of programs, descriptions of user-accessible databases used to direct the activities of programs, and for user interfaces (e.g. commands, application program interfaces) as applicable to the TOE; the evaluator would also ensure that the processor instruction set is described.

ADV_FSP.1.3C ***The functional specification shall completely represent the TSF.***

1:ADV_FSP.1-5 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

515 In order to assess the completeness of the TSF representation, the evaluator consults the user guidance and the administrator guidance. None of these should describe security functionality that is absent from the TSF presentation of the functional specification.

4.7.2.4 Action ADV_FSP.1.2E

1:ADV_FSP.1-6 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.

516 To ensure that all SFRs are covered by the functional specification, the evaluator may construct a map between the SFRs and the functional specification. Such a map might be already provided by the developer as

evidence for meeting the correspondence (Representation correspondence (ADV_RCR).*) requirements, in which case the evaluator need only verify the completeness of this mapping, ensuring that all SFRs are mapped onto applicable TSFI presentations in the functional specification.

1:ADV_FSP.1-7 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

517 For each interface to the TSF with specific characteristics, the detailed information in the functional specification must be consistent with the SFRs. For example, if the SFRs specify through FIA_SOS.1 Verification of secrets that the password length must be eight, the TOE must have eight-character passwords.

518 For each interface in the functional specification that operates on a controlled resource, the evaluator determines whether it returns an error code that indicates a possible failure due to enforcement of one of the SFRs; if no error code is returned, the evaluator determines whether an error code should be returned. For example, an operating system might present an interface to OPEN a controlled object. The description of this interface may include an error code that indicates that access was not authorised to the object. If such an error code does not exist, the evaluator should confirm whether this is appropriate (because, perhaps, access mediation is performed on READs and WRITEs, rather than on OPENs).

4.7.2.5 Action ADV_FSP.1.3E

1:ADV_FSP.1-8 The evaluator ***shall examine*** the functional specification to determine that it is consistent with the TOE summary specification.

519 The evaluator is reminded that the TOE summary specification may be at a much higher level of abstraction than the functional specification.

4.7.3 Evaluation of Representation correspondence (ADV_RCR.1)

4.7.3.1 Objectives

520 The objective of this sub-activity is to determine whether the developer has correctly and completely implemented the requirements of the ST in the functional specification.

4.7.3.2 Input

521 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the correspondence analysis between the TOE summary specification and the functional specification.

4.7.3.3 Action ADV_RCR.1.1E

1:ADV_RCR.1-1 The evaluator ***shall examine*** the correspondence analysis between the SFRs and the functional specification to determine that the functional specification is a correct and complete representation of the SFRs.

522 The evaluator's goal in this work unit is to determine that all SFRs are represented in the functional specification and that they are represented accurately.

523 The evaluator reviews the correspondence between the SFRs and the functional specification. The evaluator looks for consistency and accuracy in the correspondence. Where the correspondence analysis indicates a relationship between an SFR and one or more interface description in the functional specification, the evaluator verifies that the interface descriptions completely and accurately represent that SFR.

524 This work unit may be done in conjunction with work units ADV_FSP.1-7 and ADV_FSP.1-8.

4.8 Guidance documents activity

525 The purpose of the guidance document activity is to judge the adequacy of the documentation describing how to use the operational TOE. Such documentation includes both that aimed at trusted administrators and non-administrator users whose incorrect actions could adversely affect the security of the TOE, as well as that aimed at untrusted users whose incorrect actions could adversely affect the security of their own data.

4.8.1 Application notes

526 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.

4.8.2 Evaluation of Administrator guidance (AGD_ADM.1)

4.8.2.1 Objectives

527 The objective of this sub-activity is to determine whether the administrator guidance describes how to administer the TOE in a secure manner.

4.8.2.2 Application notes

528 The term “administrator” is used to indicate a human user who is trusted to perform security critical operations within the TOE, such as setting TOE configuration parameters. The operations may affect the enforcement of the TSP, and the administrator therefore possesses specific privileges necessary to perform those operations. The role of the administrator(s) has to be clearly distinguished from the role of non-administrative users of the TOE.

529 There may be different administrator roles or groups defined in the ST that are recognised by the TOE and that can interact with the TSF such as auditor, administrator, or daily-management. Each role can encompass an extensive set of capabilities, or can be a single one. The capabilities of these roles and their associated privileges are described in the FMT class. Different administrator roles and groups should be taken into consideration by the administrator guidance.

4.8.2.3 Input

530 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance;

e) the secure installation, generation, and start-up procedures;

4.8.2.4 Action AGD_ADM.1.1E

AGD_ADM.1.1C ***The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.***

1:AGD_ADM.1-1 The evaluator ***shall examine*** the administrator guidance to determine that it describes the administrative security interfaces available to the administrator of the TOE.

531 The administrator guidance should contain an overview of the security functionality that is visible at the administrator interfaces.

532 The administrator guidance should identify and describe the purpose, behaviour, and interrelationships of the administrator security interfaces.

533 For each administrator security interface, the administrator guidance should:

a) describe the method(s) by which the interface is invoked (e.g. command-line, programming-language system calls, menu selection, command button);

b) describe the parameters to be set by the administrator, their valid and default values;

c) describe the immediate TSF response, message, or code returned.

AGD_ADM.1.2C ***The administrator guidance shall describe how to administer the TOE in a secure manner.***

1:AGD_ADM.1-2 The evaluator ***shall examine*** the administrator guidance to determine that it describes how to administer the TOE in a secure manner.

534 The administrator guidance describes how to operate the TOE according to the TSP in an operational environment that meets all security objectives for the operational environment as described in the ST.

AGD_ADM.1.3C ***The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.***

1:AGD_ADM.1-3 The evaluator ***shall examine*** the administrator guidance to determine that it contains warnings about functions and privileges that should be controlled in a secure processing environment.

535 The configuration of the TOE may allow users to have dissimilar privileges to make use of the different functions of the TOE. This means that some users may be authorised to perform certain functions while other users may not be so authorised. These functions and privileges should be described by the administrator guidance.

536 The administrator guidance identifies the functions and privileges that must be controlled, the types of controls required for them, and the reasons for such controls. Warnings address expected effects, possible side effects, and possible interactions with other functions and privileges.

AGD_ADM.1.4C ***The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.***

1:AGD_ADM.1-4 The evaluator ***shall examine*** the administrator guidance to determine that it describes all security parameters under the control of the administrator indicating secure values as appropriate.

537 For each security parameter, the administrator guidance should describe the purpose of the parameter, the valid and default values of the parameter, and secure and insecure use settings of such parameters, both individually or in combination.

AGD_ADM.1.5C ***The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.***

1:AGD_ADM.1-5 The evaluator ***shall examine*** the administrator guidance to determine that it describes each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

538 All types of security-relevant events are detailed, such that an administrator knows what events may occur and what action (if any) the administrator 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 administrator intervention to maintain secure operation.

AGD_ADM.1.6C ***The administrator guidance shall describe all security objectives for the operational environment that are relevant to the administrator.***

1:AGD_ADM.1-6 The evaluator ***shall examine*** the administrator guidance to determine that it describes all security objectives for the operational environment that are relevant to the administrator.

539 The evaluator analyses the security objectives for the operational environment in the ST and compares them with the administrator guidance to ensure that all security objectives for the operational environment that are relevant to the administrator are described appropriately in the administrator guidance.

4.8.3 Evaluation of User guidance (AGD_USR.1)

4.8.3.1 Objectives

540 The objectives of this sub-activity are to determine whether the user guidance describes the security functions and interfaces provided by the TSF and whether this guidance provides instructions and guidelines for the secure use of the TOE.

4.8.3.2 Application notes

541 There may be different user roles or groups defined in the ST that are recognised by the TOE and that can interact with the TSF. The capabilities of these roles and their associated privileges are described in the FMT class. Different user roles and groups should be taken into consideration by the user guidance.

4.8.3.3 Input

542 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the user guidance;

e) the administrator guidance;

f) the secure installation, generation, and start-up procedures.

4.8.3.4 Action AGD_USR.1.1E

AGD_USR.1.1C ***The user guidance shall describe the interfaces available to the non-administrative users of the TOE.***

1:AGD_USR.1-1 The evaluator ***shall examine*** the user guidance to determine that it describes the security functions and interfaces available to the non-administrative users of the TOE.

543 The user guidance should contain an overview of the security functionality that is visible at the user interfaces.

544 The user guidance should identify and describe the purpose of the security interfaces and functions.

AGD_USR.1.2C ***The user guidance shall describe the use of the interfaces available to the non-administrative users of the TOE.***

1:AGD_USR.1-2 The evaluator ***shall examine*** the user guidance to determine that it describes the use of interfaces available to the non-administrative users of the TOE.

545 The user guidance should identify and describe the behaviour and interrelationship of the security interfaces available to the non-administrative users of the TOE.

546 If a non-administrative user of the TOE is allowed to invoke the TSF, the user guidance provides a description of the interfaces available to the user for that invocation.

547 For each 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 and their valid and default values;

c) describe the immediate TSF response, message, or code returned.

AGD_USR.1.3C ***The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.***

1:AGD_USR.1-3 The evaluator ***shall examine*** the user guidance to determine that it contains warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

548 The configuration of the TOE may allow users 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 user-accessible functions and privileges are described by the user guidance.

549 The user guidance should identify the functions and privileges that can be used, the types of commands required for them, and the reasons for such commands. The user guidance should contain warnings regarding the use of the functions and privileges that must be controlled. Warnings should address expected effects, possible side effects, and possible interactions with other functions and privileges.

AGD_USR.1.4C ***The user guidance shall describe all security objectives for the operational environment that are relevant to the user.***

1:AGD_USR.1-4 The evaluator ***shall examine*** the user guidance to determine that it describes all security objectives for the operational environment that are relevant to the user.

550 The evaluator analyses the security objectives for the operational environment in the ST and compares them with the user guidance to ensure that all security objectives for the operational environment that are relevant to the user are described appropriately in the user guidance.

551 The user guidance should provide advice regarding effective use of the TSF (e.g. reviewing password composition practices, suggested frequency of user file backups, discussion on the effects of changing user access privileges).

4.9 Tests activity

552 The goal of this activity is to determine, by independently testing a subset of the TSF, whether the TOE behaves as specified in the functional specification.

4.9.1 Application notes

553 The size and composition of the evaluator's test subset depends upon several factors discussed in the independent testing (ATE_IND.1 Independent testing - conformance) sub-activity. One such factor affecting the composition of the subset is *known public domain weaknesses*, information to which the evaluator needs access (e.g. from a scheme).

554 To create tests, the evaluator needs to understand the desired expected behaviour of a security function in the context of the requirements it is to satisfy. The evaluator may choose to focus on one security function of the TSF at a time, examining the ST requirement and the relevant parts of the functional specification and guidance documentation to gain an understanding of the way the TOE is expected to behave.

4.9.2 Evaluation of Independent testing (ATE_IND.1)

4.9.2.1 Objectives

555 The goal of this activity is to determine, by independently testing a subset of the TSF, whether the TOE behaves as specified in the functional specification.

4.9.2.2 Input

556 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance;

e) the secure installation, generation, and start-up procedures;

f) the TOE suitable for testing.

4.9.2.3 Action ATE_IND.1.1E

ATE_IND.1.1C ***The TOE shall be suitable for testing.***

1: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.

557 The TOE referred to in the developer's test plan should have the same unique reference as established in the ST introduction.

558 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.

559 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.

560 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.

1:ATE_IND.1-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state.

561 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 ADO_IGS.1

Installation, generation, and start-up procedures 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, generate and start up the TOE, using the supplied guidance only.

562 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 ADO_IGS.1-2.

4.9.2.4 Action ATE_IND.1.2E

1:ATE_IND.1-3 The evaluator ***shall devise*** a test subset.

563 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.

564 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.

565 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 rigourously 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.

566 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) Known public domain weaknesses commonly associated with the type of TOE (e.g. operating system, firewall). Know public domain weaknesses associated with the type of TOE will influence the selection process of the test subset. The evaluator should include those interfaces that are associated with known public domain weaknesses for that type of TOE in the subset (known public domain weaknesses in this context does not refer to vulnerabilities as such but to inadequacies or problem areas that have been experienced with this particular type of TOE).

b) Significance of interfaces. Those interfaces more significant than others should be included in the test subset. An input to this determination could be the number of SFRs mapping to this interface (as determined in Representation correspondence (ADV_RCR)).

c) Complexity of the interface. Complex interfaces may require complex tests that impose onerous requirements on the developer or evaluator, which will 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.

d) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and their inclusion in the subset may maximize 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.

e) 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.

f) 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, the corresponding interfaces should be strong candidates for testing.

567 This guidance articulates factors to consider during the selection process of an appropriate test subset, but these are by no means exhaustive.

1: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.

568 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);

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 a security function (e.g. packet generators) or make observations of a security function (e.g. network analysers).

569 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.

570 The evaluator's test documentation should specify the derivation of each test, tracing it back to the relevant interface(s).

1:ATE_IND.1-5 The evaluator ***shall conduct*** testing.

571 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.

1: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.

572 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.

573 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.

1:ATE_IND.1-7 The evaluator ***shall check*** that all actual test results are consistent with the expected test results.

574 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.

1:ATE_IND.1-8 The evaluator ***shall report*** in the ETR the evaluator testing effort, outlining the testing approach, configuration, depth and results.

575 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.

576 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.

577 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.

5 EAL4 evaluation

5.1 Introduction

578 EAL4 provides a moderate to high level of assurance. The security functions are analysed using a functional specification, guidance documentation, the high-level and low-level design of the TOE, and a subset of the implementation to understand the security behaviour. The analysis is supported by independent testing of a subset of the SFRs, evidence of developer testing based on the functional specification and the high level design, selective confirmation of the developer test results, analysis of strengths of the functions, evidence of a developer search for vulnerabilities, and an independent vulnerability analysis demonstrating resistance to low attack potential penetration attackers. Further assurance is gained through the use of an informal model of the TOE security policy and through the use of development environment controls, automated TOE configuration management, and evidence of secure delivery procedures.

5.2 Objectives

579 The objective of this chapter is to define the minimal evaluation effort for achieving an EAL4 evaluation and to provide guidance on ways and means of accomplishing the evaluation.

5.3 EAL4 evaluation relationships

580 An EAL4 evaluation covers the following:

a) evaluation input task (Chapter 2);

b) EAL4 evaluation activities comprising the following:

1) evaluation of the ST (Section 5.4);

2) evaluation of the configuration management (Section 5.5);

3) evaluation of the delivery and operation documents (Section 5.6);

4) evaluation of the development documents (Section 5.7);

5) evaluation of the guidance documents (Section 5.8);

6) evaluation of the life cycle support (Section 5.9);

7) evaluation of the tests (Section 5.10);

8) testing (Section 5.10);

9) evaluation of the vulnerability assessment (Section 5.11);

c) evaluation output task (Chapter 2).

581 The evaluation activities are derived from the EAL4 assurance requirements contained in CC Part 3.

582 The ST evaluation is started prior to any TOE evaluation sub-activities since the ST provides the basis and context to perform these sub-activities.

583 The sub-activities comprising an EAL4 evaluation are described in this chapter. Although the sub-activities can, in general, be started more or less coincidentally, some dependencies between sub-activities have to be considered by the evaluator.

5.4 Security Target evaluation activity

584 This section 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.

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

5.4.1 Application notes

5.4.1.1 ST evaluation relationships

586 The activities to conduct a complete ST evaluation cover the following:

a) evaluation input task (Section 2);

b) ST evaluation activity, comprising the following sub-activities:

1) evaluation of the ST introduction (Section 5.4.4);

2) evaluation of the conformance claims (Section 5.4.2);

3) evaluation of the security problem definition (Section 5.4.7);

4) evaluation of the security objectives (Section 5.4.5);

5) evaluation of the extended security requirements 5.4.3);

6) evaluation of the security requirements (Section 5.4.6);

7) evaluation of the TOE summary specification (Section 5.4.8).

c) evaluation output task (Section 2).

587 The evaluation input and evaluation output tasks are described in Section 2. The evaluation activities are derived from the ASE assurance requirements contained in CC Part 3.

588 The sub-activities comprising an ST evaluation are described in this clause. Although the sub-activities can, in general, be started more or less coincidentally, some dependencies between sub-activities have to be considered by the evaluator.

589 Some of the information required for the ST may be included by reference. For example if compliance to a PP is claimed, some information in the PP such as the threats may be included by reference only. All material that is referred to in such a way is considered to be part of the ST and should conform to the ASE criteria.

5.4.1.2 Re-using the evaluation results of certified PPs

590 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.

591 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.

592 An example would be where the PP contained 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.

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

5.4.2 Evaluation of Conformance claims (ASE_CCL.1)

5.4.2.1 Objectives

594 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.

5.4.2.2 Input

595 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.

5.4.2.3 Action ASE_CCL.1.1E

4: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.

596 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.

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

4: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.

4: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.

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

598 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.

4: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.

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

600 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.

4: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.

601 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).

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

4: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.

603 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).

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

4:ASE_CCL.1-8 The evaluator ***shall check*** that the conformance claim states a claim of either package-name conformant or package-name augmented.

605 If the package conformance claim contains package-name conformant, the evaluator determines that the ST contains no security requirements in addition to those included in the package.

606 If the package conformance claim contains package-name augmented, the evaluator determines that the ST includes at least one security requirement in addition to those included in the package.

4: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.

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

608 The relation between the types could be simple: a firewall ST claiming conformance to a firewall PP, or more complex: a smartcard ST claiming conformance to a number of PPs at the same time: a PP for the integrated circuit, a PP for the smartcard OS, and two PPs for two applications on the smartcard.

4: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.

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

610 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP. In this instance the statement of SPD must be stated in exactly the same wording as that used in the PP. The ST may repeat any threats, OSPs and/or assumptions or it may include them by reference to the PP they come from.

611 Where strict or demonstrable conformance is required by the PP, the conformance claim rationale should provide a tracing between the statement of SDP in the ST and that in the PP. This tracing should be sufficient for the evaluator to determine that all threats, assumptions and OSPs detailed in the PP are represented in the ST.

612 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add threats, OSPs and/or assumptions to those drawn from those in the PPs.

4: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.

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

614 The conformance claim rationale will be trivial in the case where exact conformance is required by the PP. In this instance the security objectives must be stated in exactly the same wording as that used in the PP. The ST may repeat any security objective, or it may include it by reference to the PP it comes from.

615 Where strict or demonstrable conformance is required by the PP, the conformance claim rationale should provide a tracing between the statement of security objectives in the ST and that in the PP. This tracing should be sufficient for the evaluator to determine that all security objectives detailed in the PP are represented in the ST.

616 The evaluator is reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add objectives to those drawn from those in the PPs.

4: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.

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

618 The ST may repeat any security requirements or it may include them by reference to the PP(s) they come from. If, however, the PP security requirements include uncompleted operations, or the ST author has applied the refinement operation on any PP security requirements, then these security requirements must be fully present in the ST.

619 For exact conformance, the conformance rationale will be trivial, as the statement of security requirements in the ST must include the same requirements as in the PPs, with no additions, deletions or substitutions.

620 For strict conformance, the conformance rationale will be trivial again; demonstrating that the statement of requirements in the ST is a non-strict super set of those in the PP. That is, that all requirements in the PP have been included in the ST, possibly with some additional requirements.

621 For demonstrable conformance, the evaluator determines that the justification for the security requirements in the PP demonstrates that each requirement is represented by one or more security requirements in the ST.

622 The evaluator is also reminded that if strict or demonstrable conformance with PPs is required, the ST author is allowed to add security requirements to those drawn from those PPs.

4:ASE_CCL.1-13 The evaluator ***shall examine*** the conformance claim rationale to determine that that the completion of the security requirements in the ST are consistent, in the manner specified in the PP, with those in the PP.

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

624 The PP may already have partially completed operations in a requirement, or set other limits on the completion of those operations. If this is the case, the evaluator determines that the corresponding requirement in the ST is completed consistent with these partial completions and/or limits.

625 An example of an inconsistent completion is a PP that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "The TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "The TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The ST that claims conformance to this PP, copies the requirement in the ST and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

626 Note that, if the PP in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the ST with any number other than 5 would be an inconsistent completion.

4:ASE_CCL.1-14 The evaluator ***shall examine*** the ST to determine that it is consistent with all security requirements in the packages for which conformance is being claimed.

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

628 The ST may repeat any security requirements or it may include them by reference to the package(s) they come from. If, however, the package security requirements include uncompleted operations, or the ST author has applied the refinement operation on any package security requirements, then these security requirements must be fully present in the ST.

629 The evaluator is also reminded that if the conformance claim is package-name augmented the ST author is permitted to add security requirements to those drawn from that package.

4:ASE_CCL.1-15 The evaluator ***shall examine*** the ST to determine that all security requirements in the ST that were taken from a security requirements package or PP are completed consistently with that security requirements package or PP.

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

631 If the security requirements package has already partially completed operations in a requirement, or has set other limits on the completion of those operations, the evaluator determines that the corresponding requirement in the ST is completed consistent with these partial completions and/or limits.

632 An example of an inconsistent completion is a package that partially completes the first assignment in FIA_AFL.1 Authentication failure handling "The TSF shall detect when [assignment: number] unsuccessful authentication attempts occur..." as "The TSF shall detect when [assignment: a number between 1 and 5] unsuccessful authentication attempts occur...". The ST that claims conformance to this package, copies the requirement in the ST and completes it as "The TSF shall detect when 8 unsuccessful authentication attempts occur...".

633 Note that, if the security requirements package in the example above would mandate exactly 5 unsuccessful authentication attempts, a completion in the ST with any number other than 5 would be an inconsistent completion.

5.4.3 Evaluation of Extended components definition (ASE_ECD.1)

5.4.3.1 Objectives

634 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 could not have been clearly expressed using existing CC Part 2 or CC Part 3 components.

5.4.3.2 Input

635 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.3.3 Action ASE_ECD.1.1E

4: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 Part 3.

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

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

637 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.

4: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.

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

639 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

640 If the extended component is a member of an existing CC Part 2 or 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.

641 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.

642 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

643 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.

644 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.

4: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.

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

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

4:ASE_ECD.1-5 The evaluator ***shall examine*** the extended components definition to determine that each definition of an extended functional component identifies all applicable audit information of that component.

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

648 The evaluator confirms that no applicable security relevant events that are candidates for audit have been overlooked by the ST author.

649 For guidance on audit information of a component, see CC Part 2, Section 2.1.2.5

4:ASE_ECD.1-6 The evaluator ***shall examine*** the extended security requirement components definition to determine that each definition of an extended functional component identifies all applicable security management information of that component.

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

651 The evaluator confirms that no applicable security management functions for this component have been overlooked by the ST author.

652 For guidance on security management information of a component, see CC Part 2, Section 2.1.2.4

4:ASE_ECD.1-7 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.

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

654 The evaluator determines that the extended functional component is consistent with CC Part 2 Section 2.1.3 Component structure.

655 If the extended functional component uses operations, the evaluator determines that the extended functional component is consistent with CC Part 1 Section 4.4.1.3 Component.

656 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 2.2.1 Component changes highlighting.

4:ASE_ECD.1-8 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.

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

658 The evaluator determines that all new functional families are defined consistent with CC Part 2 Section 2.1.2 Family structure.

4:ASE_ECD.1-9 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.

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

660 The evaluator determines that all new functional classes are defined consistent with CC Part 2 Section 2.1.1 Class structure.

4:ASE_ECD.1-10 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.

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

662 The evaluator determines that the extended assurance component definition is consistent with CC Part 3 Section 2.1.3 Assurance component structure.

663 If the extended assurance component uses operations, the evaluator determines that the extended assurance component is consistent with CC Part 1 Section 4.4.1.3 Component.

664 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 2.1.3 Assurance component structure.

4:ASE_ECD.1-11 The evaluator ***shall examine*** the extended components definition to determine that for each defined extended assurance component, applicable methodology has been provided.

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

666 The evaluator determines that for each evaluator action element of each extended SAR one or more work units is provided and that succesfully performing all work units for a given evaluator action element will demonstrate that the element has been achieved.

4:ASE_ECD.1-12 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.

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

668 The evaluator determines that all new assurance families are defined consistent with CC Part 3 Section 2.1.2 Assurance family structure.

4:ASE_ECD.1-13 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.

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

670 The evaluator determines that all new assurance classes are defined consistent with CC Part 3 Section 2.1.1 Class structure.

4:ASE_ECD.1-14 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 compliance or noncompliance can be demonstrated.

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

672 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.

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

674 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 requirements are to be used as a model for determining what constitutes compliance with this requirement.

5.4.3.4 Action ASE_ECD.1.2E

4:ASE_ECD.1-15 The evaluator ***shall examine*** the extended components definition to determine that each extended component can not be clearly expressed using existing components.

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

676 The evaluator determines that each extended component cannot be clearly expressed using existing components. The evaluator should take components from CC Part 2 and 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.

677 The evaluator is reminded that the role of this work unit is to preclude unnecessary duplication of components, that is, components that can be clearly expressed 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 with existing components.

5.4.4 Evaluation of ST introduction (ASE_INT.1)

5.4.4.1 Objectives

678 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.

5.4.4.2 Input

679 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.4.3 Action ASE_INT.1.1E

4: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.

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

680 The evaluator determines that the ST reference identifies the ST itself, so that it can 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.

681 In evaluations where a CM system is provided, the evaluator could 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).

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

682 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.

683 This work unit is limited to the TOE reference in the ST, checking whether the TOE is actually labelled with this reference, and whether these references are consistent, is covered by the ACM_CAP CM capabilities family.

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

684 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 are evaluated, yet the TOE reference does not reflect this, are not allowed.

4: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.

685 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.

686 The evaluator determines whether 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.

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

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

687 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.

688 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 can not operate in such an operational environment, the evaluator determines that the TOE overview adequately discusses this.

4: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.

689 While some TOEs can 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 is therefore considered to be satisfied.

690 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 should be 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.

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

691 The evaluator determines that the TOE description discusses the hardware, firmware and software components and/or modules that constitute the TOE at a level of detail that is sufficient to give the reader a general understanding of those components and/or modules.

692 The evaluator also determines that the TOE description lists all guidance that is part of the TOE.

693 The evaluator also determines that the TOE description describes exactly where the boundary lies between the TOE hardware/software/firmware and any non-TOE hardware/software/firmware.

694 The evaluator also determines that the TOE description describes exactly where the boundary between the TOE guidance and any non-TOE guidance lies.

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

695 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.

696 The evaluator also determines that the TOE description describes exactly where the boundary lies between functionality provided by the TOE, and functionality provided by any non-TOE hardware/software/firmware.

5.4.4.4 Action ASE_INT.1.2E

4: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.

5.4.5 Evaluation of Security objectives (ASE_OBJ.1)

5.4.5.1 Objectives

697 The objective of this sub-activity is to determine whether the security objectives adequately and completely address the security problem definition, that the division of this problem between the TOE, its

development environment, and its operational environment is clearly defined, and whether the security objectives are internally consistent.

5.4.5.2 Input

698 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.5.3 Action ASE_OBJ.1.1E

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

4:ASE_OBJ.1-1 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the TOE.

699 The evaluator checks that the security objectives for the TOE are identified, and that they are clearly separated from the security objectives for the development environment and the security objectives for the operational environment.

ASE_OBJ.1.2C ***The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs met by that security objective.***

4:ASE_OBJ.1-2 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 organisational policies met by the objectives.

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

701 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.1.3C ***The statement of security objectives shall describe the security objectives for the development environment.***

4:ASE_OBJ.1-3 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the development environment

702 The evaluator checks that the security objectives for the development environment are identified, and that they are also clearly separated from the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.1.4C ***The security objectives rationale shall trace each security objective for the development environment back to threats countered by that security objective and OSPs met by that security objective.***

4:ASE_OBJ.1-4 The evaluator ***shall check*** that the security objectives rationale traces the security objectives for the development environment back to threats countered by that security objective and OSPs met by that security objective.

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

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

ASE_OBJ.1.5C ***The statement of security objectives shall describe the security objectives for the operational environment***

4:ASE_OBJ.1-5 The evaluator ***shall check*** that the statement of security objectives defines the security objectives for the operational environment.

705 The evaluator checks that the security objectives for the operational environment are identified, and that they are also clearly separated from the security objectives for the TOE and the security objectives for the development environment.

ASE_OBJ.1.6C ***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.***

4:ASE_OBJ.1-6 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.

706 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.

707 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.1.7C ***The security objectives rationale shall demonstrate that the security objectives counter all threats.***

4:ASE_OBJ.1-7 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.

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

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

710 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 threats are sufficiently mitigated.

711 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 could be as minimal as “Security Objective X directly counters threat Y”.

712 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.1.8C ***The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.***

4:ASE_OBJ.1-8 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.

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

714 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 implemented.

715 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 implementation of the OSP.

716 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 could be as minimal as “Security Objective X directly enforces OSP Y”.

ASE_OBJ.1.9C ***The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.***

4:ASE_OBJ.1-9 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.

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

718 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 is consistent with the assumption.

719 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 achieving consistency with the assumption.

720 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 could be as minimal as “Security Objective for the Operational Environment X directly upholds Assumption Y”.

5.4.5.4 Action ASE_OBJ.1.2E

4:ASE_OBJ.1-10 The evaluator ***shall examine*** the statement of security objectives to determine that it is internally consistent.

721 The evaluator should compare the security objectives with each other to determine whether they contradict each other, or whether there may be conditions in which they contradict each other.

722 Examples of such contradictions are:

a) “a user's identity shall never be released” and “actions of a user shall be logged with that user's identity”.

b) “the network connection in the operational environment shall be 100% available” and “the network connection in the operational environment shall fail in a secure manner by shutting down its services gracefully”.

c) “it shall not be possible for type X users to access type Y data”, “type X users shall be able to export type Y data out of the TOE” may contradict unless the type Y data is protected in another way.

5.4.6 Evaluation of Security requirements (ASE_REQ.2)

5.4.6.1 Objectives

723 The objective of this sub-activity is to determine whether the SFRs and SARs are clear, unambiguous and canonically formulated, whether they are internally consistent, and whether they meet the security objectives of the TOE and the security objectives for the development environment.

5.4.6.2 Input

724 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.6.3 Action ASE_REQ.2.1E

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

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

725 The evaluator determines that all SFRs are 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 compliant with;

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

e) by reproduction in the ST.

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

727 If an SFR is reproduced in the ST, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 2.

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

728 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 compliant with;

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

e) by reproduction in the ST.

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

730 If an SAR is reproduced in the ST, the evaluator determines that it has been reproduced correctly by comparing it to the definition of its component in CC Part 3.

ASE_REQ.2.2C ***The statement of security requirements shall identify all operations on the security requirements.***

4:ASE_REQ.2-3 The evaluator ***shall check*** that the statement of security requirements identifies all operations on the security requirements.

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

ASE_REQ.2.3C ***All assignment and selection operations shall be completed.***

4:ASE_REQ.2-4 The evaluator ***shall examine*** the statement of security requirements to determine that each assignment and each selection operation is completed.

732 The evaluator determines that there are no choices left in the assignments and selections of all SFRs and all SARs.

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

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

733 An assignment operation is only allowed where specifically permitted in a component.

734 The evaluator compares each assignment with the component from which it is derived to determine that the values of the parameters or variables chosen comply with the indicated type required by the assignment. An assignment may only be completed with “None” if this is specifically allowed.

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

735 The evaluator determines that each iteration of a requirement is different from each other iteration of that requirement (at least one element is different), or that the requirement applies to a different part of the TOE.

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

736 A selection operation is only allowed where specifically permitted in a component.

737 The evaluator compares each selection with the component from which it is derived to determine that the selected item or items are one or more of the items indicated within the selection portion of the component. The evaluator also determines that where a selection explicitly states “choose one of”, only one item is selected.

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

738 The evaluator determines for each refinement that the component is refined in such manner that a TOE meeting the refined requirement also meets the unrefined requirement. If the refined requirement exceeds this boundary it is considered to be an extended requirement.

739 A special case of refinement is an editorial refinement, where a small change is made in a requirement, i.e. rephrasing a sentence due to adherence to proper English grammar. This change is not allowed to modify the meaning of the requirement in any way. The evaluator is reminded that editorial refinements have to be clearly identified.

740 Another special case of refinement is where multiple iterations of the same requirement are used, each with different refinements, where some of the refined iterations do not meet the full scope of the original requirement. This is acceptable, provided that all iterations of the refined requirement taken collectively, meet the entire scope of the original requirement.

741 In addition, a refinement should be related to the original requirement. Refining an audit requirement with an extra element on prevention of electromagnetic radiation is normally not allowed. This refinement should be added to another requirement, or if no applicable requirement to refine can be found, be formulated as an extended requirement.

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.***

4: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.

742 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.

743 A justification that a dependency is not met can 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.***

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

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

745 Failure to trace implies that either the security requirements rationale is incomplete, the security objectives for the TOE are incomplete, or that 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.***

4: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.

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

747 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.

748 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.

749 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.

750 The evaluator takes into account that the TOE should usually have some form of protection for itself, otherwise it will not be able to uphold its security objectives. After all, if the TSF itself can be corrupted, it will not perform its duties for long in a hostile environment.

751 While examining the justification, the evaluator takes into account that SFRs can be bypassed, tampered with, deactivated, or attacked without being detected, and that this may lead to the security objectives for the TOE not being achieved. In particular, the evaluator closely examines cases where:

a) Reference mediation (FPT_RVM) is not included, as this indicates possible bypass;

b) Domain separation (FPT_SEP) is not included, as this indicates possible logical tampering;

c) TSF physical protection (FPT_PHP) is not included, as this indicates possible physical tampering;

d) FAU: Security audit components are not included, as this indicates that attacks can be performed without being detected;

e) FMT: Security management components have been included, as this provides a possibility to modify the behaviour of other SFRs,

752 and these cases are not or not sufficiently addressed by the security objectives for the operational environment.

ASE_REQ.2.8C ***The security requirements rationale shall trace each SAR back to the security objectives for the development environment.***

4:ASE_REQ.2-12 The evaluator ***shall check*** that the security requirements rationale traces each SAR back to the security objectives for the development environment.

753 The evaluator determines that each SAR is traced back to at least one security objective for the development environment.

754 Failure to trace implies that either the security requirements rationale is incomplete, the security objectives for the development environment are incomplete, or that the SAR has no useful purpose.

4:ASE_REQ.2-13 The evaluator ***shall examine*** the security requirements rationale to determine that for each security objective for the development environment it justifies that the SARs are suitable to meet that security objective for the development environment.

755 If no SARs trace back to the security objective for the development environment, this work unit fails.

756 The evaluator determines that the justification for a security objective for the development environment demonstrates that the SARs are sufficient: if all SARs that trace back to the objective are satisfied, the security objective for the development environment is achieved.

757 The evaluator also determines that each SAR that traces back to a security objective for the development environment is necessary, when the SAR is satisfied, it actually contributes to achieving the security objective.

758 Note that the tracings from SARs to security objectives for the development environment provided in the security requirements rationale may be a part of the justification, but do not constitute a justification by themselves.

5.4.6.4 Action ASE_REQ.2.2E

4:ASE_REQ.2-14 The evaluator ***shall examine*** the statement of security requirements to determine that it is internally consistent.

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

760 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 “all objects”, “all subjects” etc., that these requirements do not conflict.

761 Some possible conflicts are:

a) FRU_RSA.2 Minimum and maximum quotas specifying a minimum number of resources available to a user and FTA_MCS.1 Basic limitation on multiple concurrent sessions specifying a maximum number of sessions available for a user. If the resources are somehow linked to sessions, these requirements may conflict;

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

c) FPR_ANO.1 Anonymity, FAU_GEN.1 Audit data generation specifying that subject identity is to be logged, and FAU_SAR.1 Audit review specifying who can read the audit records. If people from whom the activities of users should be hidden, can read the audit logs of these activities, these requirements may conflict;

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

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

5.4.7 Evaluation of Security problem definition (ASE_SPD.1)

5.4.7.1 Objectives

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

5.4.7.2 Input

763 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.7.3 Action ASE_SPD.1.1E

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

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

764 If all security objectives are derived from assumptions and 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.

765 The evaluator determines that the security problem definition describes the threats that must be countered by the TOE, its development environment, its operational environment or combinations of these three.

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

4: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.

766 If all security objectives are derived from assumptions and 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.

767 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.***

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

768 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.

769 The evaluator determines that OSP statements are made in terms of rules, practices or guidelines that must be followed by the TOE, its development environment, its operational environment or combinations of these three.

770 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.***

4: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.

771 If the threats and/or OSPs already sufficiently address the physical, personnel, and connectivity aspects of the TOE, this work unit is not applicable and is therefore considered to be satisfied.

772 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.

5.4.8 Evaluation of TOE summary specification (ASE_TSS.1)

5.4.8.1 Objectives

773 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.

5.4.8.2 Input

774 The evaluation evidence for this sub-activity is:

a) the ST.

5.4.8.3 Action ASE_TSS.1.1E

4:ASE_TSS.1-1 The evaluator ***shall examine*** the TOE summary specification to determine that it describes how the TOE meets each SFR.

775 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.

776 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.

5.4.8.4 Action ASE_TSS.1.2E

4: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.

777 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.

5.5 Configuration management activity

778 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.

5.5.1 Evaluation of CM automation (ACM_AUT.1)

5.5.1.1 Objectives

779 The objective of this sub-activity is to determine whether changes to the implementation representation are controlled with the support of automated tools, thus making the CM system less susceptible to human error or negligence.

5.5.1.2 Input

780 The evaluation evidence for this sub-activity is:

a) the configuration management documentation.

5.5.1.3 Action ACM_AUT.1.1E

ACM_AUT.1.1C ***The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.***

4:ACM_AUT.1-1 The evaluator ***shall check*** the CM plan for a description of the automated measures to control access to the TOE implementation representation.

4:ACM_AUT.1-2 The evaluator ***shall examine*** the automated access control measures to determine that they are effective in preventing unauthorised modification of the TOE implementation representation.

781 The evaluator reviews the configuration management documentation to identify those individuals or roles authorised to make changes to the TOE implementation representation. For example, once it is under configuration management, access to an element of the implementation representation may only be allowed for the individual who performs the software integration role.

782 The evaluator should exercise the automated access control measures to determine whether they can be bypassed by an unauthorised role or user. This determination need only comprise a few basic tests.

ACM_AUT.1.2C ***The CM system shall provide an automated means to support the generation of the TOE.***

4:ACM_AUT.1-3 The evaluator ***shall check*** the CM documentation for automated means to support generation of the TOE from its implementation representation.

783 In this work unit the term “generation” applies to those processes adopted by the developer to progress the TOE from its implementation to a state ready to be delivered to the end customer.

784 The evaluator should verify the existence of automated generation support procedures within the CM documentation.

4:ACM_AUT.1-4 The evaluator ***shall examine*** the automated generation procedures to determine that they can be used to support generation of the TOE.

785 The evaluator determines that by following the generation procedures a TOE would be generated that reflects its implementation representation. The customer can then be confident that the version of the TOE delivered for installation implements the SFRs as described in the ST. For example, in a software TOE this may include checking that the automated generation procedures help to ensure that all source files and related libraries that are part of the TSF are included in the compiled object code.

786 It should be noted that this requirement is only to provide support. For example, an approach that placed Unix makefiles under configuration management should be sufficient to meet the aim, given that in such a case automation would have made a significant contribution to accurate generation of the TOE. Automated procedures can assist in identifying the correct configuration items to be used in generating the TOE.

ACM_AUT.1.3C ***The CM plan shall describe the automated tools used in the CM system.***

4:ACM_AUT.1-5 The evaluator ***shall check*** that the CM plan includes information on the automated tools used in the CM system.

ACM_AUT.1.4C ***The CM plan shall describe how the automated tools are used in the CM system.***

4:ACM_AUT.1-6 The evaluator ***shall examine*** the information relating to the automated tools provided in the CM plan to determine that it describes how they are used.

787 The information provided in the CM plan provides the necessary detail for a user of the CM system to be able to operate the automated tools correctly in order to maintain the integrity of the TOE. For example, the information provided may include a description of:

a) the functionality provided by the tools;

b) how this functionality is used by the developer to control changes to the implementation representation;

c) how this functionality is used by the developer to support generation of the TOE.

5.5.1.4 Implied evaluator action

ACM_AUT.1.1D

4:ACM_AUT.1-7 The evaluator ***shall examine*** the CM system to determine that the automated tools and procedures described in the CM plan are used.

788 This work unit may be viewed as an additional activity to be carried out in parallel with the evaluator's examination into the use of the CM system required by CM capabilities (ACM_CAP). The evaluator looks for evidence that the tools and procedures are in use. This should include a visit to the development site to witness operation of the tools and procedures, and an examination of evidence produced through their use.

5.5.2 Evaluation of CM capabilities (ACM_CAP.4)

5.5.2.1 Objectives

789 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.

5.5.2.2 Input

790 The evaluation evidence for this sub-activity is:

a) the ST;

b) the TOE suitable for testing;

c) the configuration management documentation.

5.5.2.3 Action ACM_CAP.4.1E

ACM_CAP.4.1C ***The reference for the TOE shall be unique to each version of the TOE.***

4:ACM_CAP.4-1 The evaluator ***shall check*** that the version of the TOE provided for evaluation is uniquely referenced.

791 The evaluator should use the developer's CM system to validate the uniqueness of the reference by checking the configuration list to ensure that the configuration items are uniquely identified. Evidence that the version provided for evaluation is uniquely referenced may be incomplete if only one version is examined during the evaluation, and the evaluator should look for a referencing system that is capable of supporting unique references (e.g. use of numbers, letters or dates). However, the absence of any reference will normally lead to a fail verdict against this requirement unless the evaluator is confident that the TOE can be uniquely identifed.

792 The evaluator should seek to examine more than one version of the TOE (e.g. during rework following discovery of a vulnerability), to check that the two versions are referenced differently.

ACM_CAP.4.2C ***The TOE shall be labelled with its reference.***

4:ACM_CAP.4-2 The evaluator ***shall check*** that the TOE provided for evaluation is labelled with its reference.

793 The evaluator should ensure that the TOE contains a unique reference such that it is possible to distinguish different versions of the TOE. 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).

794 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.

4:ACM_CAP.4-3 The evaluator ***shall check*** that the TOE references used are consistent.

795 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. The evaluator can use the configuration list that is part of the provided CM documentation to verify the consistent use of identifiers.

796 The evaluator also verifies that the TOE reference is consistent with the ST.

ACM_CAP.4.3C ***The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.***

4:ACM_CAP.4-4 The evaluator ***shall check*** that the CM documentation provided includes a configuration list.

797 A configuration list identifies the items being maintained under configuration control.

4:ACM_CAP.4-5 The evaluator ***shall check*** that the CM documentation provided includes a CM plan.

4:ACM_CAP.4-6 The evaluator ***shall check*** that the CM documentation provided includes an acceptance plan.

ACM_CAP.4.4C ***The configuration list shall uniquely identify all configuration items that comprise the TOE.***

4:ACM_CAP.4-7 The evaluator ***shall check*** that the configuration list uniquely identifies each configuration item.

798 The configuration list contains a list of the configuration items that comprise the TOE, together with 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.

ACM_CAP.4.5C ***The configuration list shall describe the configuration items that comprise the TOE.***

4:ACM_CAP.4-8 The evaluator ***shall examine*** the configuration list to determine that it identifies the configuration items that comprise the TOE.

799 The minimum scope of configuration items to be covered in the configuration list is given by CM scope (ACM_SCP).

ACM_CAP.4.6C ***The CM documentation shall describe the method used to uniquely identify the configuration items.***

4:ACM_CAP.4-9 The evaluator ***shall examine*** the method of identifying configuration items to determine that it describes how configuration items are uniquely identified.

ACM_CAP.4.7C ***The CM system shall uniquely identify all configuration items.***

4:ACM_CAP.4-10 The evaluator ***shall examine*** the configuration items to determine that they are identified in a way that is consistent with the CM documentation.

800 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.

ACM_CAP.4.8C ***The CM plan shall describe how the CM system is used.***

4:ACM_CAP.4-11 The evaluator ***shall examine*** the CM plan to determine that it describes how the CM system is used to maintain the integrity of the TOE configuration items.

801 The descriptions contained in a CM plan may include:

a) all activities performed in the TOE development environment that are subject to configuration management procedures (e.g. creation, modification or deletion of a configuration item);

b) the roles and responsibilities of individuals required to perform operations on individual configuration items (different roles may be

identified for different types of configuration item (e.g. design documentation or source code));

c) the procedures that are used to ensure that only authorised individuals can make changes to configuration items;

d) the procedures that are used to ensure that concurrency problems do not occur as a result of simultaneous changes to configuration items;

e) 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;

f) 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).

ACM_CAP.4.9C ***The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.***

4:ACM_CAP.4-12 The evaluator ***shall check*** the CM documentation to ascertain that it includes the CM system records identified by the CM plan.

802 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 ACM_CAP.4.10C. Example output could include change control forms, or configuration item access approval forms.

4:ACM_CAP.4-13 The evaluator ***shall examine*** the evidence to determine that the CM system is being used as it is described in the CM plan.

803 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.

804 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 practice 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.

805 It is expected that the evaluator will visit the development site in support of this activity.

ACM_CAP.4.10C ***The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.***

4:ACM_CAP.4-14 The evaluator ***shall check*** that the configuration items identified in the configuration list are being maintained by the CM system.

806 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. high-level design 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.

ACM_CAP.4.11C ***The CM system shall provide measures such that only authorised changes are made to the configuration items.***

4:ACM_CAP.4-15 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.

807 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 and already examined as part of the work unit ACM_CAP.4-13. The evaluator may also witness a demonstration of the CM system to ensure that the access control measures employed are operating effectively.

808 The developer will have provided automated access control measures as part of the CM system and as such their suitability may be verified under the component ACM_AUT.1 Partial CM automation.

ACM_CAP.4.12C ***The CM system shall support the generation of the TOE.***

4:ACM_CAP.4-16 The evaluator ***shall check*** the CM documentation for procedures for supporting the generation of the TOE.

809 In this work unit the term “generation” applies to those processes adopted by the developer to progress the TOE from implementation to a state acceptable for delivery to the end customer.

810 The evaluator verifies the existence of generation support procedures within the CM documentation. The generation support procedures provided by the developer may be automated, and as such their existence may be verified under the component ACM_AUT.1.2C.

4:ACM_CAP.4-17 The evaluator ***shall examine*** the TOE generation procedures to determine that they are effective in helping to ensure that the correct configuration items are used to generate the TOE.

811 The evaluator determines that by following the generation support procedures the version of the TOE expected by the customer (i.e. as described in the TOE ST and consisting of the correct configuration items) would be generated and delivered for installation at the customer site. For example, in a software TOE this may include checking that the procedures ensure that all source files and related libraries are included in the compiled object code.

812 The evaluator should bear in mind that the CM system need not necessarily possess the capability to generate the TOE, but should provide support for the process that will help reduce the probability of human error.

ACM_CAP.4.13C ***The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.***

4:ACM_CAP.4-18 The evaluator ***shall examine*** the acceptance procedures to determine that they describe the acceptance criteria to be applied to newly created or modified configuration items.

813 An acceptance plan describes the procedures that are to be used to ensure that the constituent parts of the TOE are of adequate quality prior to incorporation into the TOE. The acceptance plan should identify the acceptance procedures to be applied:

a) at each stage of the construction of the TOE (e.g. module, integration, system);

b) to the acceptance of software, firmware and hardware components;

c) to the acceptance of previously evaluated components.

814 The description of the acceptance criteria may include identification of:

a) developer roles or individuals responsible for accepting such configuration items;

b) any acceptance criteria to be applied before the configuration items are accepted (e.g. successful document review, or successful testing in the case of software, firmware or hardware).

5.5.3 Evaluation of CM scope (ACM_SCP.2)

5.5.3.1 Objectives

815 The objective of this sub-activity is to determine whether the developer performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, the CM documentation and security flaws.

5.5.3.2 Input

816 The evaluation evidence for this sub-activity is:

a) the configuration item list.

5.5.3.3 Action ACM_SCP.2.1E

ACM_SCP.2.1C ***The list of configuration items shall include the following: implementation representation, security flaws, and the evaluation evidence required by the SARs in the ST.***

4:ACM_SCP.2-1 The evaluator ***shall check*** that the configuration item list includes the set of items required by the CC.

817 The list includes the following:

a) the TOE implementation representation (i.e., the components or subsystems that compose the TOE). For a software-only TOE, the implementation representation may consist solely of source code; for a TOE that includes a hardware platform, the implementation representation may refer to a combination of software, firmware and a description of the hardware.

b) the evaluation evidence required by the SARs in the ST.

c) 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).

5.6 Delivery and operation activity

818 The purpose of the delivery and operation activity is to judge the adequacy of the documentation of the procedures used to ensure that the TOE is installed, generated, and started in the same way the developer intended it to be and that it is delivered without modification. This includes both the procedures taken while the TOE is in transit, as well as the initialisation, generation, and start-up procedures.

5.6.1 Evaluation of Delivery (ADO_DEL.2)

5.6.1.1 Objectives

819 The objective of this sub-activity is to determine whether the delivery documentation describes all procedures used to maintain security and detect modification or substitution of the TOE when distributing the TOE to the user's site.

5.6.1.2 Input

820 The evaluation evidence for this sub-activity is:

a) the delivery documentation.

5.6.1.3 Action ADO_DEL.2.1E

ADO_DEL.2.1C ***The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.***

4:ADO_DEL.2-1 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 user's site.

821 Interpretation of the term “necessary” will need to consider the nature of the TOE and information contained in the ST. The level of protection provided should be commensurate with the other SARs. In some cases these may not be explicitly expressed in relation to delivery. The evaluator should determine that a balanced approach has been taken, such that delivery does not present an obvious weak point in an otherwise secure development process.

822 The delivery procedures describe proper procedures to determine the identification of the TOE and to maintain security of the TOE during transfer of the TOE or its component parts. The procedures describe which parts of the TOE need to be covered by these procedures. It should contain procedures for physical or electronic (e.g. for downloading off the Internet) distribution where applicable. The delivery procedures refer to the entire TOE, including applicable software, hardware, firmware and documentation.

823 The emphasis in the delivery documentation is likely to be on measures related to integrity, as technical measures are required to be applied to maintain integrity during the TOE delivery. 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.

824 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).

825 Standard commercial practice for packaging and delivery may be acceptable. This includes shrink wrapped packaging, a security tape or a sealed envelope. For the distribution, the public mail or a private distribution service may be acceptable.

826 The suitability of the choice of the delivery procedures is influenced by the TOE (e.g. whether it is software or hardware) and by the security objectives. In cases where the delivery procedures differ for different parts of the TOE, the totality of procedures should be considered.

ADO_DEL.2.2C ***The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.***

4:ADO_DEL.2-2 The evaluator ***shall examine*** the delivery documentation to determine that it describes how the various procedures and technical measures provide for the detection of modifications or any discrepancy between the developer's master copy and the version received at the user site.

827 Checksum procedures, software signature, or tamper proof seals may be used by the developer to ensure that tampering can be detected. The developer may also employ other procedures (e.g. a recorded delivery service) that register the name of the originator and supplie the name to the receiver.

828 Technical measures for the detection of any discrepancy between the developer's master copy and the version received at the user site should be described in the delivery procedures.

ADO_DEL.2.3C ***The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.***

4:ADO_DEL.2-3 The evaluator ***shall examine*** the delivery documentation to determine that it describes how the various mechanisms and procedures allow detection of attempted masquerading even in cases in which the developer has sent nothing to the user's site.

829 This requirement may be fulfilled by delivering the TOE or parts of it (e.g. by an agent known to and trusted by both developer and user). For a software TOE a digital signature may be appropriate.

830 If the TOE is delivered by electronic download, the security can be maintained by using digital signatures, integrity checksums, or encryption.

5.6.1.4 Implied evaluator action

ADO_DEL.2.2D

4:ADO_DEL.2-4 The evaluator ***shall examine*** aspects of the delivery process to determine that the delivery procedures are used.

831 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 should seek some assurance that they are applied in practice. 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 at the user's site (e.g. checking for tamper proof seals);

c) observing that the process is applied in practice when the evaluator obtains the TOE through regular channels;

d) questioning end users as to how the TOE was delivered.

832 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.

5.6.2 Evaluation of Installation, generation and start-up (ADO_IGS.1)

5.6.2.1 Objectives

833 The objective of this sub-activity is to determine whether the procedures and steps for the secure installation, generation, and start-up of the TOE have been documented and result in a secure configuration.

5.6.2.2 Application notes

834 The installation, generation, and start-up procedures refer to all installation, generation, and start-up procedures, regardless of whether they are performed at the user's site or at the development site that are necessary to progress the TOE to the secure configuration as described in the ST.

5.6.2.3 Input

835 The evaluation evidence for this sub-activity is:

a) the administrator guidance;

b) the secure installation, generation, and start-up procedures;

c) the TOE suitable for testing.

5.6.2.4 Action ADO_IGS.1.1E

4:ADO_IGS.1-1 The evaluator ***shall check*** that the procedures necessary for the secure installation, generation and start-up of the TOE have been provided.

836 If it is not anticipated that the installation, generation, and start-up procedures will or can be reapplied (e.g. because the TOE may already be delivered in an operational state) this work unit (or the effected parts of it) is not applicable, and is therefore considered to be satisfied.

5.6.2.5 Action ADO_IGS.1.2E

4:ADO_IGS.1-2 The evaluator ***shall examine*** the provided installation, generation, and start-up procedures to determine that they describe the steps necessary for secure installation, generation, and start-up of the TOE.

837 If it is not anticipated that the installation, generation, and start-up procedures will or can be reapplied (e.g. because the TOE may already be delivered in an operational state) this work unit (or the effected parts of it) is not applicable, and is therefore considered to be satisfied.

838 The installation, generation, and start-up procedures may provide detailed information about the following:

a) changing the installation specific security characteristics of entities under the control of the TSF;

b) handling exceptions and problems;

c) minimum system requirements for secure installation if applicable.

839 In order to confirm that the installation, generation, and start-up procedures result in a secure configuration, the evaluator may follow the developer's procedures and may perform the activities that customers are usually expected to perform to install, generate, and start-up the TOE (if applicable to the TOE), using the supplied guidance documentation only. This work unit might be performed in conjunction with the ATE_IND.1-2 work unit.

5.7 Development activity

840 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. 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 external interfaces of the TSF), a high-level design (which describes the architecture of the TSF in terms of internal subsystems), and a low-level design (which describes the architecture of the TSF in terms of internal modules). Additionally, there is an implementation description (a source code level description), a security policy model (which describes the security policies enforced by the TSF) and a representation correspondence (which

maps representations of the TSF to one another in order to ensure consistency).

5.7.1 Application notes

841 The CC requirements for design documentation are levelled by formality. The CC considers a document's degree of formality (that is, whether it is informal, semiformal or formal) 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.

842 An informal functional specification comprises a description of the purpose and method-of-use of externally-visible 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.

843 An informal high-level design is expressed in terms of sequences of actions that occur in each subsystem in response to stimulus at its interface. 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 high-level 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.

844 An informal low-level design is expressed in terms of sequences of actions that occur in a module in response to stimulus at its interface. For example, a virtual private networking subsystem might be composed of modules that create session keys, that encrypt traffic, that decrypt traffic, and that decide whether traffic needs to be encrypted. The low-level description of the encryption module would describe the steps that the module takes when it receives a traffic stream that is to be encrypted.

845 While the functional specification describes the functionality and services, the model describes the policies that that functionality and those services enforce. An informal model is simply a description of the security policies enforced by services or functionality available at the external interface. For example, access control policies would describe the resources being protected and the conditions that must be met for access to be granted; audit policies would describe the TOE's auditable events, identifying both those that are selectable by the administrator and those that are always audited;

identification and authentication policies would describe how users are identified, how those claimed identities are authenticated, and any rules affecting how identities are authenticated (e.g. users on the corporate intranet need no authentication, while external users are authenticated with one-time passwords).

846 Informality of the demonstration of correspondence need not be in a prose form; a simple two-dimensional mapping may be sufficient. For example, a matrix with modules listed along one axis and subsystems listed along the other, with the cells identifying the correspondence of the two, would serve to provide an adequate informal correspondence between the high-level design and the low-level design.

5.7.2 Evaluation of Functional specification (ADV_FSP.2)

5.7.2.1 Objectives

847 The objective of this sub-activity is to determine whether the developer has provided an adequate description of the TSF and whether this shows that all SFRs have been sufficiently addressed.

5.7.2.2 Input

848 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance.

5.7.2.3 Action ADV_FSP.2.1E

ADV_FSP.2.1C ***The functional specification shall describe the TSF and its external interfaces using an informal style.***

4:ADV_FSP.2-1 The evaluator ***shall examine*** the functional specification to determine that it contains all necessary informal explanatory text.

849 If the entire functional specification is informal, this work unit is not applicable and is therefore considered to be satisfied.

850 Supporting narrative descriptions are necessary for those portions of the functional specification that are difficult to understand only from the semiformal or formal description (for example, to make clear the meaning of any formal notation).

ADV_FSP.2.2C ***The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.***

4:ADV_FSP.2-2 The evaluator ***shall examine*** the functional specification to determine that it identifies all of the external TSF interfaces.

851 The term *external* refers to that which is visible to the user. External interfaces to the TOE are either direct interfaces to the TSF or interfaces to non-TSF portions of the TOE. However, these non-TSF interfaces might have eventual access to the TSF. These external interfaces that directly or indirectly access the TSF collectively make up the TSF interface (TSFI). Figure 7 shows a TOE with TSF (shaded) portions and non-TSF (empty) portions. This TOE has three external interfaces: interface *c* is a direct interface to the TSF; interface *b* is an indirect interface to the TSF; and interface *a* is an interface to non-TSF portions of the TOE. Therefore, interfaces *b* and *c* make up the TFSI.

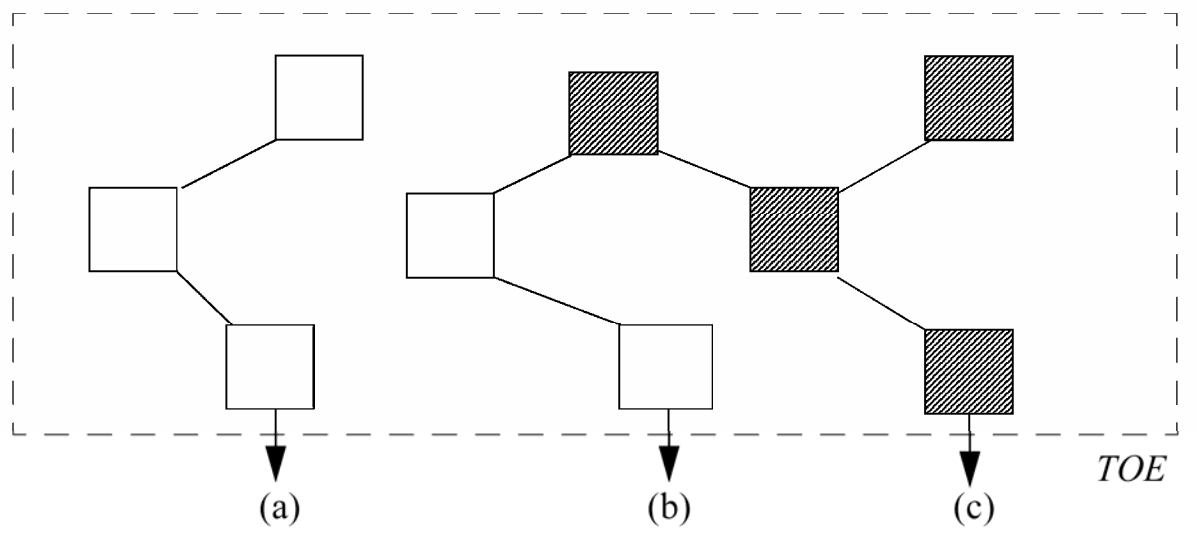

Figure 7 - TSF Interfaces

852 It should be noted that all SFRs will have some sort of externally-visible manifestation, including those SFRs whose failure is the only externally visible observation (e.g. FDP_RIP.1 Subset residual information protection, FPT_SEP.1 TSF domain separation and FPT_RVM.1 Non-bypassability of the TSP). While not all of these are necessarily interfaces from which the SFR can be tested, they are all externally-visible to some extent and must therefore be included in the functional specification.

4:ADV_FSP.2-3 The evaluator ***shall examine*** the functional specification to determine that it describes all of the external TSF interfaces.

853 For a TOE that has no threat of malicious users (i.e. TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) are rightfully excluded from its ST), the only interfaces that are described in the functional specification (and expanded upon in the other TSF representation descriptions) are those to and from the TSF. The absence of TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) presumes there is no concern for any sort

of bypassing of the security features; therefore, there is no concern with any possible impact that other interfaces might have on the TSF.

854 On the other hand, if the TOE has a threat of malicious users or bypass (i.e. TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) are included in its ST), all external interfaces are described in the functional specification, but only to the extent that the effect of each is made clear: interfaces to the security functions (i.e. interfaces b and c in Figure **7**) are completely described, while other interfaces are described only to the extent that it is clear that the TSF is inaccessible through the interface (i.e. that the interface is of type a, rather than b in Figure **7**). The inclusion of TSF physical protection (FPT_PHP), Reference mediation (FPT_RVM), and Domain separation (FPT_SEP) implies a concern that all interfaces might have some effect upon the TSF. Because each external interface is a potential TSF interface, the functional specification must contain a description of each interface in sufficient detail so that an evaluator can determine whether the interface is security relevant.

855 Some architectures lend themselves to readily provide this interface description in sufficient detail for groups of external interfaces. For example, a kernel architecture is such that all calls to the operating system are handled by kernel programs; any calls that might violate the TSP must be called by a program with the privilege to do so. All programs that execute with privilege must be included in the functional specification. Any program external to the kernel that executes without privilege is incapable of violating the TSP (i.e. such programs are interfaces of type *a*, rather than *b* in Figure 7) and may, therefore, be excluded from the functional specification. It is worth noting that, while the evaluator's understanding of the interface description can be expedited in cases where there is a kernel architecture, such an architecture is not necessary.

4:ADV_FSP.2-4 The evaluator ***shall examine*** the presentation of the TSFI to determine that it adequately and correctly describes the complete behaviour of the TSF at each external interface describing effects, exceptions and error messages.

856 In order to assess the adequacy and correctness of an interface's presentation, the evaluator uses the functional specification and the user and administrator guidance to assess the following factors:

a) All security relevant user input parameters (or a characterisation of those parameters) should be identified. For completeness, parameters outside of direct user control should be identified if they are usable by administrators.

b) Complete security relevant behaviour described in the reviewed guidance should be reflected in the description of semantics in the functional specification. This should include an identification of the behaviour in terms of events and the effect of each event. For example, if an operating system provides a rich file system interface, where it provides a different error code for each reason why a file is not opened upon request, the functional specification should explain

that a file is either opened upon request, or else that the request is denied, along with a listing of the reasons why the open request might be denied (e.g. access denied, no such file, file is in use by another user, user is not authorised to open the file after 5pm, etc.). It would be insufficient for the functional specification merely to explain that a file is either opened upon request, or else that an error code is returned. The description of the semantics should include how the security requirements apply to the interface (e.g. whether the use of the interface is an auditable event and, if so, the information that can be recorded).

c) All interfaces are described for all possible modes of operation. If the TSF provides the notion of privilege, the description of the interface should explain how the interface behaves in the presence or absence of privilege.

d) The information contained in the descriptions of the security relevant parameters and syntax of the interface should be consistent across all documentation.

857 Verification of the above is done by reviewing the SFRs, the functional specification as well as the user and administrator guidance provided by the developer. For example, if the TOE were an operating system and its underlying hardware, the evaluator would look for discussions of user-accessible programs, descriptions of protocols used to direct the activities of programs, descriptions of user-accessible databases used to direct the activities of programs, and for user interfaces (e.g. commands, application program interfaces) as applicable to the TOE; the evaluator would also ensure that the processor instruction set is described.

858 This review might be iterative, such that the evaluator would not discover the functional specification to be incomplete until the high-level design, source code, or other evidence is examined and found to contain parameters or error messages that have been omitted from the functional specification.

ADV_FSP.2.3C ***The functional specification shall completely represent the TSF.***

4:ADV_FSP.2-5 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

859 In order to assess the completeness of the TSF representation, the evaluator consults the user guidance, and the administrator guidance. None of these should describe security funtionality that are absent from the TSF presentation of the functional specification.

860 This review might be iterative, such that the evaluator would not discover the TSF to be incompletely represented until the high-level design, source code, or other evidence is examined and found to contain functionality that had been omitted from the functional specification.

ADV_FSP.2.4C ***The functional specification shall include rationale that the TSF is completely represented.***

4:ADV_FSP.2-6 The evaluator ***shall examine*** the functional specification to determine that it contains a convincing argument that the TSF is completely represented by the functional specification.

861 The evaluator determines that there is a convincing argument that there are no interfaces in the TSFI that are missing from the functional specification. This may include a description of the procedure or methodology that the developer used to ensure that all external interfaces to the TSF are covered. The argument would prove inadequate if, for example, the evaluator discovers commands, parameters, error messages, or other interfaces to the TSF in other evaluation evidence, yet absent from the functional specification.

5.7.2.4 Action ADV_FSP.2.2E

4:ADV_FSP.2-7 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs

862 To ensure that all SFRs are covered by the functional specification, the evaluator may construct a map between the SFRs and the functional specification. Such a map might be already provided by the developer as evidence for meeting the correspondence (ADV_RCR.*) requirements, in which case the evaluator need only verify the completeness of this mapping, ensuring that all SFRs are mapped onto applicable TSFI presentations in the functional specification.

4:ADV_FSP.2-8 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.

863 For each interface to the TSF with specific characteristics, the detailed information in the functional specification must be consistent with the SFRs. For example, if the SFRs specify through FIA_SOS.1 Verification of secrets that the password length must be eight characters, the TOE must have eight-character passwords.

864 For each interface in the functional specification that operates on a controlled object, the evaluator determines whether it returns an error code that indicates a possible failure due to enforcement of one or more of the SFRs; if no error code is returned, the evaluator determines whether an error code should be returned. For example, an operating system might present an interface to OPEN a controlled object. The description of this interface may include an error code that indicates that access was not authorised to the object. If such an error code does not exist, the evaluator should confirm whether this is appropriate (because, perhaps, access mediation is performed on READs and WRITEs, rather than on OPENs).

5.7.2.5 Action ADV_FSP.2.3E

4:ADV_FSP.2-9 The evaluator ***shall examine*** the functional specification to determine that it is consistent with the TOE summary specification.

865 The evaluator is reminded that the TOE summary specification may be at a much higher level of abstraction than the functional specification.

5.7.3 Evaluation of High-level design (ADV_HLD.2)

5.7.3.1 Objectives

866 The objective of this sub-activity is to determine whether the high-level design provides a description of the TSF in terms of major structural units (i.e. subsystems), provides a description of the interfaces to these structural units, and is a correct realisation of the functional specification.

5.7.3.2 Input

867 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design.

5.7.3.3 Action ADV_HLD.2.1E

ADV_HLD.2.1C ***The presentation of the high-level design shall be informal.***

4:ADV_HLD.2-1 The evaluator ***shall examine*** the high-level design to determine that it contains all necessary informal explanatory text.

868 If the entire high-level design is informal, this work unit is not applicable and is therefore considered to be satisfied.

869 Supporting narrative descriptions are necessary for those portions of the high-level design that are difficult to understand only from the semiformal or formal description (for example, to make clear the meaning of any formal notation).

ADV_HLD.2.2C ***The high-level design shall describe the structure of the TSF in terms of subsystems.***

4:ADV_HLD.2-2 The evaluator ***shall examine*** the high-level design to determine that the TSF is described in terms of subsystems.

870 With respect to the high-level design, the term subsystem refers to large, related units (such as memory-management, file-management, process-management). Breaking a design into the basic functional areas aids in the understanding of the design.

871 The primary purpose for examining the high-level design is to aid the evaluator's understanding of the TSF. The developer's choice of subsystem definition are an important aspect of making the high-level design useful in understanding the TSF's intended operation. As part of this work unit, the evaluator should make an assessment as to the appropriateness of the number and nature of subsystems presented by the developer. The evaluator should ensure that the decomposition of the TSF into subsystems is sufficient for the evaluator to gain a high-level understanding of how the functionality of the TSF is provided.

872 The subsystems used to describe the high-level design need not be called “subsystems”, but should represent a similar level of decomposition. For example, the design may be decomposed using “layers” or “managers”.

873 There may be some interaction between the choice of subsystem definition and the scope of the evaluator's analysis. A discussion on this interaction is found following work unit ADV_HLD.2-10.

ADV_HLD.2.3C ***The high-level design shall describe the security functionality provided by each subsystem of the TSF.***

4:ADV_HLD.2-3 The evaluator ***shall examine*** the high-level design to determine that it describes the security functionality of each subsystem.

874 The security functionality of a subsystem is a description of what the subsystem does. This should include a description of any actions that the subsystem may be directed to perform and the effects the subsystem may have on the security state of the TOE (e.g. changes in subjects, objects, security databases).

ADV_HLD.2.4C ***The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.***

4:ADV_HLD.2-4 The evaluator ***shall check*** the high-level design to determine that it identifies all hardware, firmware, and software required by the TSF.

875 If the ST contains no IT-related security objectives for the operational environment, this work unit is not applicable and is therefore considered to be satisfied.

876 The evaluator determines whether the list of hardware, firmware, or software required by the TSF as stated in the high-level design is consistent with the IT-related security objectives for the operational environment.

4:ADV_HLD.2-5 The evaluator ***shall examine*** the high-level design to determine that it includes a presentation of the security functionality provided by the supporting protection mechanisms implemented in the underlying hardware, firmware, or software.

877 If the ST contains no IT-related security objectives for the operational environment, this work unit is not applicable and is therefore considered to be satisfied.

878 The presentation of the security functionality provided by the underlying abstract machine on which the TOE executes need not be at the same level of detail as the presentation of TSF subsystems in the high-level design. The presentation should explain how the TSF uses this functionality to support the TSF meeting the SFRs.

879 The IT-related security objectives for the operational environment may be abstract, particularly if they are intended to be capable of being satisfied by a variety of different combinations of hardware, firmware, or software.

ADV_HLD.2.5C ***The high-level design shall identify all interfaces to the subsystems of the TSF.***

4:ADV_HLD.2-6 The evaluator ***shall check*** that the high-level design identifies the interfaces to the TSF subsystems.

880 The high-level design includes, for each subsystem, the name of each of its interfaces.

ADV_HLD.2.6C ***The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.***

4:ADV_HLD.2-7 The evaluator ***shall check*** that the high-level design identifies which of the interfaces to the subsystems of the TSF are externally visible.

881 As discussed under work unit ADV_FSP.1-3, external interfaces (i.e. those visible to the user) may directly or indirectly access the TSF. Any external interface that accesses the TSF either directly or indirectly is included in the identification for this work unit. External interfaces that do not access the TSF need not be included.

ADV_HLD.2.7C ***The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.***

4:ADV_HLD.2-8 The evaluator ***shall examine*** the high-level design to determine that it describes the interfaces to each subsystem in terms of their purpose and method of use, and provides details of effects, exceptions and error messages, as appropriate.

882 The high-level design should include descriptions in terms of the purpose and method of use for all interfaces of each subsystem. Such descriptions may be provided in general terms for some interfaces, and in more detail for others. In determining the level of detail of effects, exceptions and error messages that should be provided, the evaluator should consider the purposes of this analysis and the uses made of the interface by the TOE. For example, the evaluator needs to understand the nature of the interactions between

subsystems to establish confidence that the TOE design is sound, and may be able to obtain this understanding with only a general description of some of the interfaces between subsystems. In particular, internal subsystem interfaces that are not called by any other subsystem would not normally require detailed descriptions.

883 The level of detail may also depend on the testing approach adopted to meet the Depth (ATE_DPT) requirement. For example, a different amount of detail may be needed for a testing approach that tests only through external interfaces than one that tests through both external and internal subsystem interfaces.

884 Detailed descriptions would include details of any input and output parameters, of the effects of the interface, and of any exceptions or error messages it produces. In the case of external interfaces, the required description is probably included in the functional specification and may be referenced in the high-level design without replication.

ADV_HLD.2.8C ***The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.***

4:ADV_HLD.2-9 The evaluator ***shall check*** that the high-level design describes the separation of the TOE into TSP-enforcing and other subsystems.

885 The TSF comprises all the parts of the TOE that have to be relied upon for enforcement of the TSP. Because the TSF includes both subsystems that directly enforce the TSP, and also those subsystems that, while not directly enforcing the TSP, contribute to the enforcement of the TSP in a more indirect manner, all TSP-enforcing subsystems are contained in the TSF. Subsystems that play no role in TSP enforcement are not part of the TSF. An entire subsystem is part of the TSF if any portion of it is.

886 As explained under work unit ADV_HLD.2-3, the developer's choice of subsystem definition is an important aspect of making the high-level design useful in understanding the TOE's intended operation. However, the choice of subsystems also affects the scope of the TSF, because a subsystem with any function that directly or indirectly enforces the TSP is part of the TSF. While the goal of understandability is important, it is also helpful to limit the extent of the TSF so as to reduce the amount of analysis that is required. The two goals of understandability and scope reduction may sometimes work against each other. The evaluator should bear this in mind when assessing the choice of subsystem definition.

5.7.3.4 Action ADV_HLD.2.2E

4:ADV_HLD.2-10 The evaluator ***shall examine*** the high-level design to determine that it is an accurate instantiation of the SFRs.

887 The evaluator validates the subsystem interface specifications by ensuring that:

a) the interface specifications are consistent with the description of the purpose of the subsystem;

b) the interface specifications are consistent with their use by other subsystems;

c) the interrelationships between subsystems that are needed in order that each TSP-enforcing function is correctly supported are correctly stated.

4:ADV_HLD.2-11 The evaluator ***shall examine*** the high-level design to determine that it is a complete instantiation of the SFRs.

888 The evaluator ensures that all SFRs are mapped onto applicable sections of the high-level design. This determination should be made in conjunction with the ADV_RCR.1 Informal correspondence demonstration sub-activity.

889 The evaluator analyses the high-level design to determine that each SFR is completely described by the subsystem specifications, and that there are no subsystems on which an SFR relies for which there is no specification in the high-level design.

5.7.4 Evaluation of Implementation representation (ADV_IMP.1)

5.7.4.1 Objectives

890 The objective of this sub-activity is to determine whether the implementation representation is sufficient to satisfy the functional requirements of the ST and is a correct realisation of the low-level design.

5.7.4.2 Input

891 The evaluation evidence for this sub-activity is:

a) the ST;

b) the low-level design;

c) the subset of the implementation representation.

5.7.4.3 Action ADV_IMP.1.1E

ADV_IMP.1.1C ***The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.***

4:ADV_IMP.1-1 The evaluator ***shall examine*** the implementation representation to determine that it unambiguously defines the TSF to a level of detail such that the TSF can be generated without any further design decisions.

892 This work unit requires the evaluator to confirm that the implementation representation is suitable for analysis. The evaluator should consider the

process needed to generate the TSF from the representation provided. If the process is well-defined, requiring no further design decisions (for example, requiring only the compilation of source code, or the building of hardware from hardware drawings), then the implementation representation can be said to be suitable.

893 Any programming languages used must be well defined with an unambiguous definition of all statements, as well as the compiler options used to generate the object code. This determination will have been made as part of the ALC_TAT.1 Well-defined development tools sub-activity.

4:ADV_IMP.1-2 The evaluator ***shall examine*** the implementation representation provided by the developer to determine that it is sufficiently representative.

894 The developer is required to provide the implementation representation for only a subset of the TSF. The developer can select and offer an initial subset, but the evaluator may require additional portions, or even different subsets.

895 The evaluator determines the adequacy and appropriateness of the subset by applying the principles of sampling.

896 In determining the appropriateness of the subset, the evaluator decides if it is suitable for use in aiding the evaluator to understand and gain assurance of the correctness of the implementation of the TSF. In making this determination, the evaluator should consider the different methods of representation used by the developer, so that the evaluator is satisfied that a representative subset has been selected.

897 For example, if some of the implementation representation is known to have originated from different development organisations, the selected subset should contain samples from each of the different creating organisations. If the implementation representation source code includes different forms of programming languages, the subset should contain samples of each different language.

898 In the case that the implementation representation includes hardware drawings, several different portions of the TSF should be included in the subset. For example, for a TSF including a desktop computer, the selected subset should contain samples for peripheral controllers as well as the main computer board.

899 Other factors that might influence the determination of the subset include:

a) the complexity of the design (if the design complexity varies across the TSF, the subset should include some portions with high complexity);

b) scheme requirements;

c) the results of other design analysis sub-activities (such as work units related to the low-level or high-level design) that might indicate

portions of the TSF in which there is a potential for ambiguity in the design; and

d) the evaluator's judgement as to portions of the implementation representation that might be useful for the evaluator's independent vulnerability analysis (sub-activity AVA_VLA.2 Independent vulnerability analysis).

5.7.4.4 Action ADV_IMP.1.2E

4:ADV_IMP.1-3 The evaluator ***shall examine*** the implementation representation subset to determine that it accurately instantiates those SFRs relevant to the subset.

900 The evaluator may make use of the low-level design to assess if the portions in the implementation representation subset, in combination with other portions as described in the low-level design, work together to instantiate the SFRs.

5.7.5 Evaluation of Low-level design (ADV_LLD.1)

5.7.5.1 Objectives

901 The objective of this sub-activity is to determine whether the low-level design is sufficient to satisfy the SFRs, and is a correct and effective refinement of the high-level design.

5.7.5.2 Input

902 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the low-level design.

5.7.5.3 Action ADV_LLD.1.1E

ADV_LLD.1.1C ***The presentation of the low-level design shall be informal.***

4:ADV_LLD.1-1 The evaluator ***shall examine*** the low-level design to determine that it contains all necessary informal explanatory text.

903 If the entire low-level design is informal, this work unit is not applicable and is therefore considered to be satisfied.

904 Supporting narrative descriptions are necessary for those portions of the low-level design that are difficult to understand only from the semiformal or formal description (for example, to make clear the meaning of any formal notation).

ADV_LLD.1.2C ***The low-level design shall describe the TSF in terms of modules.***

4:ADV_LLD.1-2 The evaluator ***shall check*** the low-level design to determine that it describes the TSF in terms of modules.

905 The term module is used in this family by the CC to denote a less abstract entity than a subsystem. This means that it contains more detail as to, not only the module's purpose, but also the manner in which the module achieves its purpose. Ideally, the low-level design would provide all the information needed to implement the modules described in it. The later work units in this sub-activity call for specific analysis to determine that a sufficient level of detail is included. For this work unit, it is sufficient for the evaluator to verify that each module is clearly and unambiguously identified.

ADV_LLD.1.3C ***The low-level design shall describe the purpose of each module.***

4:ADV_LLD.1-3 The evaluator ***shall examine*** the low-level design to determine that it describes the purpose of each module.

906 The low-level design contains a description of the purpose of each of its modules. These descriptions should be clear enough to convey what functions the module is expected to perform. The description should provide an overview of a module's purpose and is not intended to be at the level of detail of module interface specifications.

ADV_LLD.1.4C ***The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.***

4:ADV_LLD.1-4 The evaluator ***shall examine*** the low-level design to determine that it defines the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

907 For the purpose of this analysis, modules are viewed as interacting in two ways:

a) to provide services to one another, and

b) to cooperate in support of meeting SFRs.

908 The low-level design should include specific information on these interrelationships. For example, if a module performs calculations that depend on the results of calculations in other modules, those other modules should be listed. Further, if a module provides a service intended for other modules to use in meeting SFRs, the service should be described. It is possible that the description of the purpose of a module, as analysed in the preceding work unit, is sufficient to provide this information.

ADV_LLD.1.5C ***The low-level design shall describe how each TSP-enforcing module is provided.***

4:ADV_LLD.1-5 The evaluator ***shall examine*** the low-level design to determine that it describes how each of the TSP-enforcing modules is provided.

909 It is this description in the low-level design that is key to the assessment as to whether the low-level design is sufficiently refined to permit an implementation to be created. The evaluator should analyse the description from the point of view of an implementor. If the evaluator, using the implementor's viewpoint, is unclear on any aspect of how the module could be implemented, the description is incomplete. Note that there is no requirement that a module be implemented as a separate unit (be it a program, a subprogram, or a hardware component); but the low-level design may be sufficiently detailed to permit such an implementation.

ADV_LLD.1.6C ***The low-level design shall identify all interfaces to the modules of the TSF.***

4:ADV_LLD.1-6 The evaluator ***shall check*** that the low-level design identifies the interfaces to the TSF modules.

910 The low-level design should include, for each module, the name of each of its entry points.

ADV_LLD.1.7C ***The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.***

4:ADV_LLD.1-7 The evaluator ***shall check*** that the low-level design identifies which of the interfaces to the modules of the TSF are externally visible.

911 As discussed under work unit ADV_FSP.2-3, external interfaces (i.e. those visible to the user) may directly or indirectly access the TSF. Any external interface that accesses the TSF either directly or indirectly is included in the identification for this work unit. External interfaces that do not access the TSF need not be included.

ADV_LLD.1.8C ***The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.***

4:ADV_LLD.1-8 The evaluator ***shall examine*** the low-level design to determine that it describes the interfaces to each module in terms of their purpose and method of use, and provides details of effects, exceptions and error messages, as appropriate.

912 The module interface descriptions may be provided in general terms for some interfaces, and in more detail for others. In determining the necessary level of detail of effects, exceptions and error messages, the evaluator should consider the purposes of this analysis and the uses made of the interface by the TOE. For example, the evaluator needs to understand the general nature of the interactions between modules to establish confidence that the TOE design is sound, and may be able to obtain this understanding with only a general description of some of the interfaces between modules. In particular,

internal entry points that are not called by any other module would not normally require detailed descriptions.

913 This work unit may be performed in conjunction with the evaluator's independent vulnerability analysis, which is part of the Vulnerability analysis (AVA_VLA) sub-activity.

914 Detailed descriptions would include details of any input and output parameters, of the effects of the interface, and of any exceptions or error messages it produces. In the case of external interfaces, the required description is probably included in the functional specification and can be referenced in the low-level design without replication.

ADV_LLD.1.9C ***The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.***

4:ADV_LLD.1-9 The evaluator ***shall check*** that the low-level design describes the separation of the TOE into TSP-enforcing and other modules.

915 The TSF comprises all the parts of the TOE that have to be relied upon for enforcement of the TSP. Because the TSF includes both modules that directly enforce the TSP, and also those modules that, while not directly enforcing the TSP, contribute to the enforcement of the TSP in a more indirect manner, all TSP-enforcing modules are contained in the TSF. Modules that cannot affect TSP enforcement are not part of the TSF.

5.7.5.4 Action ADV_LLD.1.2E

4:ADV_LLD.1-10 The evaluator ***shall examine*** the low-level design to determine that it is an accurate instantiation of the SFRs.

916 The evaluator validates the module interface specifications by ensuring that:

a) the interface specifications are consistent with the description of the purpose of the module;

b) the interface specifications are consistent with their use by other modules;

c) the interrelationships between modules that are needed in order that each TSP-enforcing function is correctly supported are correctly stated.

4:ADV_LLD.1-11 The evaluator ***shall examine*** the low-level design to determine that it is a complete instantiation of the SFRs.

917 The evaluator ensures that all SFRs are mapped onto applicable sections of the low-level design. This determination should be made in conjunction with the ADV_RCR.1 Informal correspondence demonstration sub-activity.

918 The evaluator analyses the low-level design to determine that each SFR is completely described by the module specifications, and that there are no

modules on which an SFR relies for which there is no specification in the low-level design.

5.7.6 Evaluation of Representation correspondence (ADV_RCR.1)

5.7.6.1 Objectives

919 The objective of this sub-activity is to determine whether the developer has correctly and completely implemented the requirements of the ST, functional specification, high-level design and low-level design in the implementation representation.

5.7.6.2 Input

920 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the low-level design;

e) a subset of the implementation representation;

f) the correspondence analysis between the TOE summary specification and the functional specification;

g) the correspondence analysis between the functional specification and the high-level design;

h) the correspondence analysis between the high-level design and the low-level design;

i) the correspondence analysis between the low-level design and the subset of the implementation representation.

5.7.6.3 Action ADV_RCR.1.1E

4:ADV_RCR.1-1 The evaluator ***shall examine*** the correspondence analysis between the SFRs and the functional specification to determine that the functional specification is a correct and complete representation of the SFRs.

921 The evaluator's goal in this work unit is to determine that all SFRs are represented in the functional specification and that they are represented accurately.

922 The evaluator reviews the correspondence between the SFRs and the functional specification. The evaluator looks for consistency and accuracy in the correspondence. Where the correspondence analysis indicates a relationship between an SFR and one or more interface description in the

functional specification, the evaluator verifies that the interface descriptions completely and accurately represent that SFR.

923 This work unit may be done in conjunction with work units ADV_FSP.2-8 and ADV_FSP.2-9.

4:ADV_RCR.1-2 The evaluator ***shall examine*** the correspondence analysis between the functional specification and the high-level design to determine that the high-level design is a correct and complete representation of the functional specification.

924 The evaluator uses the correspondence analysis, the functional specification, and the high-level design to ensure that it is possible to map elements in the functional specification onto a TSF subsystem described in the high-level design. The evaluator verifies that the high-level design includes a description of a correct realisation of each element in the functional specifcation.

4:ADV_RCR.1-3 The evaluator ***shall examine*** the correspondence analysis between the high-level design and the low-level design to determine that the low-level design is a correct and complete representation of the high-level design.

925 The evaluator uses the correspondence analysis, the high-level design, and the low-level design to ensure that it is possible to map each TSF module identified in the low-level design onto a TSF subsystem described in the high-level design. For each TSF subsystem, the correspondence indicates which TSF modules are involved in implementing that subsystem. The evaluator verifies that the low-level design includes a description of a correct realisation of each TSF subsystem.

4:ADV_RCR.1-4 The evaluator ***shall examine*** the correspondence analysis between the low-level design and the subset of the implementation representation to determine that the subset is a correct and complete representation of those portions of the low-level design that are refined in the implementation representation.

926 Since the evaluator examines only a subset of the implementation representation, this work unit is performed by only assessing the correspondence analysis of the subset of the implementation representation to the relevant parts of the low-level design.

5.7.7 Evaluation of Security policy modeling (ADV_SPM.1)

5.7.7.1 Objectives

927 The objectives of this sub-activity are to determine whether the security policy model clearly and consistently describes the rules and characteristics of the security policies and whether this description corresponds with the functional specification.

5.7.7.2 Input

928 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the TOE security policy model;

d) the user guidance;

e) the administrator guidance.

5.7.7.3 Action ADV_SPM.1.1E

ADV_SPM.1.1C ***The TSP model shall be informal.***

4:ADV_SPM.1-1 The evaluator ***shall examine*** the security policy model to determine that it contains all necessary informal explanatory text.

929 If the entire model is informal, this work unit is not applicable and is therefore considered to be satisfied.

930 Supporting narrative descriptions are necessary for those portions of the model that are difficult to understand only from the semiformal or formal description (for example, to make clear the meaning of any formal notation).

ADV_SPM.1.2C ***The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.***

4:ADV_SPM.1-2 The evaluator ***shall check*** the model to determine that all security policies that are explicitly included in the ST are modeled.

931 The security policy is expressed by the collection of the SFRs. Therefore, to determine the nature of the security policy (and hence what policies must be modeled), the evaluator analyses the SFRs for those policies explicitly called for (by Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC), if included in the ST).

932 Depending upon the TOE, formal/semiformal modeling might not even be possible for access control. (For example, the access control policy for a firewall connected to the internet cannot be formally modeled in a useful manner because the state of the internet cannot be completely defined.). For any security policy where formal or semiformal models are not possible, the policy must be provided in an informal form.

933 If the ST contains no explicit policies (because neither Access control policy (FDP_ACC) nor Information flow control policy (FDP_IFC) are included in the ST), this work unit is not applicable and is therefore considered to be satisfied.

4:ADV_SPM.1-3 The evaluator ***shall examine*** the security policy model to determine that all security policies represented by the SFRs are modeled.

934 In addition to the explicitly-listed policies (see work unit ADV_SPM.1-2), the evaluator analyses the SFRs for those policies implied by the other functional security requirement classes. For example, inclusion of FDP requirements (other than Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC)) would need a description of the Data Protection policy being enforced; inclusion of any FIA: Identification and authentication requirements would necessitate that a description of the Identification and Authentication policies be present in the security policy model; inclusion of FAU: Security audit requirements need a description of the Audit policies; etc. While the other functional families are not typically associated with what are commonly referred to as security policies, they nevertheless do enforce security policies (e.g. non-repudiation, reference mediation, privacy, etc.) that must be included in the security policy model.

935 In cases where the model presentation is informal, all security policies can be modeled (i.e. described), and so must be included. For any security policy where formal or semiformal models are not possible, the policy must be provided in an informal form.

936 If the ST contains no such implicit policies, this work unit is not applicable and is therefore considered to be satisfied.

4:ADV_SPM.1-4 The evaluator ***shall examine*** the rules and characteristics of the model to determine that the modeled security behaviour of the TOE is clearly articulated.

937 The rules and characteristics describe the security posture of the TOE. It is likely that such a description would be contained within an evaluated and certified ST. In order to be considered a clear articulation, such a description should define the notion of security for the TOE, identify the security attributes of the entities controlled by the TOE and identify the TOE actions which change those attributes. For example, if a policy attempts to address data integrity concerns, the policy model would:

a) define the notion of integrity for that TOE;

b) identify the types of data for which the TOE would maintain integrity;

c) identify the entities that could modify that data;

d) identify the rules that potential modifiers must follow to modify data.

ADV_SPM.1.3C ***The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.***

4:ADV_SPM.1-5 The evaluator ***shall examine*** the security policy model rationale to determine that the behaviour modeled is consistent with respect to policies described by the security policies (as articulated by the SFRs).

938 In determining consistency, the evaluator verifies that the rationale shows that each rule or characteristic description in the model accurately reflects the intent of the security policies. For example, if a policy stated that access control was necessary to the granularity of a single individual, then a model describing the security behaviour of a TOE in the context of controlling groups of users would not be consistent. Likewise, if the policy stated that access control for groups of users was necessary, then a model describing the security behaviour of a TOE in the context of controlling individual users would also not be consistent.

939 Assurance is to be gained from an explicit and general statement of the policies underlying the SFRs. The assurance gained is two-fold: collecting the description of each security policy into a concise whole aids in understanding the details of the policies being enforced. Additionally, such a collected description makes it much easier to see any gaps or inconsistencies (which must be sought as part of the Security policy modeling (ADV_SPM).*.3C element), and provides a clear characterisation of secure states (sought as part of the Security policy modeling (ADV_SPM).*.2C element).

940 The requirement for an Informal Security Policy Model (ISPM) is met by a clear statement of the security policy. The need for a separate ISPM is not absolute, since for very straightforward policies, or those very clearly expressed in the ST, there may be no need for a separate ISPM. In such cases, different sections of the ST (e.g. the security objectives for the TOE, the SFRs) may combine together to provide a sufficient level of detail for the security policy. However, this is often not the case. For example, audit requirements may be spread throughout the SFRs, which may not provide a clear model of the overall policy. Unless another section of the ST (perhaps the security objectives for the TOE) pulls together the audit requirements into a cohesive whole, then having a separate ISPM would be necessary in order to allow for the detection of inconsistencies within the ST requirements that may otherwise pass undetected.

941 Where a developer claims that the ISPM requirements for some or all of the security policies are met by the ST, the evaluator needs to determine that this is the case by applying the requirements of the ADV_SPM.1 Informal TOE security policy model component: determining that the policy is clearly expressed, and that the model is consistent with the remainder of the ST. As part of the ISPM rationale, it is likely that, in cases where the developer claims that the ISPM is met entirely by the ST, that the rationale will reference the demonstrations of suitability and correspondence between portions of the ST. When evaluating this work-unit, the evaluator may draw upon the results of the ST evaluation in this area.

4:ADV_SPM.1-6 The evaluator ***shall examine*** the security policy model rationale to determine that the behaviour modeled is complete with respect to the policies described by the security policies (i.e. as articulated by the SFRs).

942 In determining completeness of this rationale, the evaluator considers the rules and characteristics of the security policy model and maps those rules

and characteristics to explicit policy statements (i.e. SFRs). The rationale should show that all policies that are required to be modeled have an associated rule or characteristic description in the security policy model.

943 Where a developer claims that the ISPM requirements for some or all of the security policies are met by the ST, the evaluator needs to determine that this is the case by applying the requirements of the ADV_SPM.1 Informal TOE security policy model component: determining that the policy is clearly expressed, and that the model is complete with respect to the remainder of the ST. When evaluating this work-unit, the evaluator may draw upon the results of the evaluation of the completeness of the various portions of the ST.

ADV_SPM.1.4C ***The demonstration of correspondence between the TSP model and the functional specification shall show that all of the external interfaces to the TSF in the functional specification are consistent and complete with respect to the TSP model.***

4:ADV_SPM.1-7 The evaluator ***shall examine*** the functional specification correspondence demonstration of the security policy model to determine that it identifies all external interfaces to the TSF described in the functional specification that implement a portion of the policy.

944 In determining completeness, the evaluator reviews the functional specification, identifies which external interfaces to the TSF directly support the security policy model and verifies that these interfaces are present in the functional specification correspondence demonstration of the security policy model.

4:ADV_SPM.1-8 The evaluator ***shall examine*** the functional specification correspondence demonstration of the security policy model to determine that the descriptions of the external interfaces to the TSF as implementing the security policy model are consistent with the security policy model.

945 To demonstrate consistency, the evaluator verifies that the functional specification correspondence shows that the description in the functional specification of the external interfaces to the TSF identified as implementing the policy described in the security policy model identify the same attributes and characteristics of the security policy model and enforce the same rules as the security policy model.

946 In cases where a security policy is enforced differently for untrusted users and administrators, the policies for each are described consistently with the respective behaviour descriptions in the user and administrator guidance. For example, the “identification and authentication” policy enforced upon remote untrusted users might be more stringent than that enforced upon administrators whose only point of access is within a physically-protected area; the differences in authentication should correspond to the differences in the descriptions of authentication within the user and administrator guidance.

5.8 Guidance documents activity

947 The purpose of the guidance document activity is to judge the adequacy of the documentation describing how to use the operational TOE. Such documentation includes both that aimed at trusted administrators and non-administrator users whose incorrect actions could adversely affect the security of the TOE, as well as that aimed at untrusted users whose incorrect actions could adversely affect the security of their own data.

5.8.1 Application notes

948 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.

5.8.2 Evaluation of Administrator guidance (AGD_ADM.1)

5.8.2.1 Objectives

949 The objective of this sub-activity is to determine whether the administrator guidance describes how to administer the TOE in a secure manner.

5.8.2.2 Application notes

950 The term “administrator” is used to indicate a human user who is trusted to perform security critical operations within the TOE, such as setting TOE configuration parameters. The operations may affect the enforcement of the TSP, and the administrator therefore possesses specific privileges necessary to perform those operations. The role of the administrator(s) has to be clearly distinguished from the role of non-administrative users of the TOE.

951 There may be different administrator roles or groups defined in the ST that are recognised by the TOE and that can interact with the TSF such as auditor, administrator, or daily-management. Each role can encompass an extensive set of capabilities, or can be a single one. The capabilities of these roles and their associated privileges are described in the FMT class. Different administrator roles and groups should be taken into consideration by the administrator guidance.

5.8.2.3 Input

952 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance;

e) the secure installation, generation, and start-up procedures;

5.8.2.4 Action AGD_ADM.1.1E

4:AGD_ADM.1-1 The evaluator ***shall examine*** the administrator guidance to determine that it describes the administrative security interfaces available to the administrator of the TOE.

953 The administrator guidance should contain an overview of the security functionality that is visible at the administrator interfaces.

954 The administrator guidance should identify and describe the purpose, behaviour, and interrelationships of the administrator security interfaces.

955 For each administrator security interface, the administrator guidance should:

a) describe the method(s) by which the interface is invoked (e.g. command-line, programming-language system calls, menu selection, command button);

b) describe the parameters to be set by the administrator, their valid and default values;

c) describe the immediate TSF response, message, or code returned.

4:AGD_ADM.1-2 The evaluator ***shall examine*** the administrator guidance to determine that it describes how to administer the TOE in a secure manner.

956 The administrator guidance describes how to operate the TOE according to the TSP in an operational environment that meets all security objectives for the operational environment as described in the ST.

4:AGD_ADM.1-3 The evaluator ***shall examine*** the administrator guidance to determine that it contains warnings about functions and privileges that should be controlled in a secure processing environment.

957 The configuration of the TOE may allow users to have dissimilar privileges to make use of the different functions of the TOE. This means that some users may be authorised to perform certain functions while other users may not be so authorised. These functions and privileges should be described by the administrator guidance.

958 The administrator guidance identifies the functions and privileges that must be controlled, the types of controls required for them, and the reasons for such controls. Warnings address expected effects, possible side effects, and possible interactions with other functions and privileges.

4:AGD_ADM.1-4 The evaluator ***shall examine*** the administrator guidance to determine that it describes all security parameters under the control of the administrator indicating secure values as appropriate.

959 For each security parameter, the administrator guidance should describe the purpose of the parameter, the valid and default values of the parameter, and

secure and insecure use settings of such parameters, both individually or in combination.

4:AGD_ADM.1-5 The evaluator ***shall examine*** the administrator guidance to determine that it describes each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

960 All types of security-relevant events are detailed, such that an administrator knows what events may occur and what action (if any) the administrator 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 administrator intervention to maintain secure operation.

4:AGD_ADM.1-6 The evaluator ***shall examine*** the administrator guidance to determine that it describes all security objectives for the operational environment that are relevant to the administrator.

961 The evaluator analyses the security objectives for the operational environment in the ST and compares them with the administrator guidance to ensure that all security objectives for the operational environment that are relevant to the administrator are described appropriately in the administrator guidance.

5.8.3 Evaluation of User guidance (AGD_USR.1)

5.8.3.1 Objectives

962 The objectives of this sub-activity are to determine whether the user guidance describes the security functions and interfaces provided by the TSF and whether this guidance provides instructions and guidelines for the secure use of the TOE.

5.8.3.2 Application notes

963 There may be different user roles or groups defined in the ST that are recognised by the TOE and that can interact with the TSF. The capabilities of these roles and their associated privileges are described in the FMT class. Different user roles and groups should be taken into consideration by the user guidance.

5.8.3.3 Input

964 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the user guidance;

e) the administrator guidance;

f) the secure installation, generation, and start-up procedures.

5.8.3.4 Action AGD_USR.1.1E

4:AGD_USR.1-1 The evaluator ***shall examine*** the user guidance to determine that it describes the security functions and interfaces available to the non-administrative users of the TOE.

965 The user guidance should contain an overview of the security functionality that is visible at the user interfaces.

966 The user guidance should identify and describe the purpose of the security interfaces and functions.

4:AGD_USR.1-2 The evaluator ***shall examine*** the user guidance to determine that it describes the use of interfaces available to the non-administrative users of the TOE.

967 The user guidance should identify and describe the behaviour and interrelationship of the security interfaces available to the non-administrative users of the TOE.

968 If a non-administrative user of the TOE is allowed to invoke the TSF, the user guidance provides a description of the interfaces available to the user for that invocation.

969 For each 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 and their valid and default values;

c) describe the immediate TSF response, message, or code returned.

4:AGD_USR.1-3 The evaluator ***shall examine*** the user guidance to determine that it contains warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

970 The configuration of the TOE may allow users 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 user-accessible functions and privileges are described by the user guidance.

971 The user guidance should identify the functions and privileges that can be used, the types of commands required for them, and the reasons for such

commands. The user guidance should contain warnings regarding the use of the functions and privileges that must be controlled. Warnings should address expected effects, possible side effects, and possible interactions with other functions and privileges.

4:AGD_USR.1-4 The evaluator ***shall examine*** the user guidance to determine that it describes all security objectives for the operational environment that are relevant to the user.

972 The evaluator analyses the security objectives for the operational environment in the ST and compares them with the user guidance to ensure that all security objectives for the operational environment that are relevant to the user are described appropriately in the user guidance.

973 The user guidance should provide advice regarding effective use of the TSF (e.g. reviewing password composition practices, suggested frequency of user file backups, discussion on the effects of changing user access privileges).

5.9 Life cycle support activity

974 The purpose of the life-cycle support activity is to determine the adequacy of the procedures the developer uses during the development and maintenance of the TOE. These procedures include the security measures used throughout TOE development, the life-cycle model used by the developer, and the tools used by the developer throughout the life-cycle of the TOE.

975 Developer security procedures are intended to protect the TOE and its associated design information from interference or disclosure. Interference in the developement 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.

976 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.

977 The use of well-defined development tools help to ensure that vulnerabilities are not inadvertently introduced during refinement.

5.9.1 Evaluation of Development security (ALC_DVS.1)

5.9.1.1 Objectives

978 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.

5.9.1.2 Input

979 The evaluation evidence for this sub-activity is:

a) the ST;

b) the development security documentation.

980 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 ACM_CAP.4 Generation support and acceptance procedures and ACM_SCP.2 Problem tracking CM coverage sub-activities). Evidence that the procedures are being applied is also required.

5.9.1.3 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.***

4: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.

981 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.

982 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.

983 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

– 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.

984 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. For example, development could occur at multiple facilities within a single building, multiple buildings at the same site, or at multiple sites. Development includes such tasks as creating multiple copies of the TOE, where applicable. This work-unit should not overlap with those for Delivery (ADO_DEL), but the evaluator should ensure that all aspects are covered by one sub-activity or the other.

985 Whereas the CM capabilities (ACM_CAP) requirements are fixed, those for 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 who have security clearance. The evaluators would then determine that such a policy had been applied under this sub-activity.

4: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.

986 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.

987 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.

988 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 (ACM_CAP) sub-activity. 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.

989 Whereas the CM capabilities (ACM_CAP) requirements are fixed, those for 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 who have security clearance. The evaluators would then determine that such a policy had been applied under this sub-activity.

ALC_DVS.1.2C ***The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.***

4:ALC_DVS.1-3 The evaluator ***shall check*** the development security documentation to determine that documentary evidence that would be produced as a result of application of the procedures has been generated.

990 Where documentary evidence is produced the evaluator inspects it to ensure compliance with procedures. Examples of the evidence produced may include entry logs and audit trails. The evaluator may choose to sample the evidence.

5.9.1.4 Action ALC_DVS.1.2E

4:ALC_DVS.1-4 The evaluator ***shall examine*** the development security documentation and associated evidence to determine that the security measures are being applied.

991 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.

992 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 overseer.

5.9.2 Evaluation of Life cycle definition (ALC_LCD.1)

5.9.2.1 Objectives

993 The objective of this sub-activity is to determine whether the developer has used a documented model of the TOE life-cycle.

5.9.2.2 Input

994 The evaluation evidence for this sub-activity is:

a) the ST;

b) the life-cycle definition documentation.

5.9.2.3 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.***

4: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.

995 A life-cycle model encompasses the procedures, tools and tehniques used to develop and maintain the TOE. The description of the life-cycle model should include information on the procedures, tools and techniques used by the developer (e.g. for design, coding, testing, bug-fixing). It should describe 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). ALC_LCD.1 Developer defined life-cycle model 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.***

4: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.

996 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 actions

covered under the ACM activity). 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.

997 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.

5.9.3 Evaluation of Tools and techniques (ALC_TAT.1)

5.9.3.1 Objectives

998 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.

5.9.3.2 Application notes

999 This work may be performed in parallel with the ADV_IMP.1 Subset of the implementation of the TSF sub-activity, specifically with regard to determining the use of features in the tools that will affect the object code (e.g. compilation options).

5.9.3.3 Input

1000 The evaluation evidence for this sub-activity is:

a) the development tool documentation;

b) the subset of the implementation representation.

5.9.3.4 Action ALC_TAT.1.1E

ALC_TAT.1.1C ***All development tools used for implementation shall be well-defined.***

4:ALC_TAT.1-1 The evaluator ***shall examine*** the development tool documentation provided to determine that all development tools are well-defined.

1001 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 the development tools shall unambiguously define the meaning of all statements used in the implementation.***

4:ALC_TAT.1-2 The evaluator ***shall examine*** the documentation of development tools to determine that it unambiguously defines the meaning of all statements used in the implementation.

1002 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 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.1 Subset of the implementation of the TSF 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.

1003 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.

1004 The critical test is whether the evaluator can understand the TOE source code when performing source code analysis covered in the Implementation representation (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.

1005 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.

1006 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.

ALC_TAT.1.3C ***The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.***

4: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.

1007 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.

1008 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).

5.10 Tests activity

1009 The purpose of this activity is to confirm that the TSF behaves as specified in the design documentation. This is accomplished by determining that the developer has tested the TSF against its functional specification and high-level design, gaining confidence in those test results by performing a sample of the developer's tests, and. by independently performing additional tests.

5.10.1 Application notes

5.10.1.1 Understanding the expected behaviour of the TOE

1010 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 the TSF by examining the functional specification, the high-level design, and the user and administrator guidance.

1011 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 the TSF being stimulated at either external or internal interfaces and its responses are observed. However, there may be cases where the TSF cannot be adequately tested at an interface (as may be the case, for instance, for residual information protection functionality); in such cases, other means will need to be employed.

5.10.1.2 Testing vs. alternate approaches to verify the expected behaviour of an interface

1012 In cases where it is impractical or inadequate to test at an interface, 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 EAL is not commensurate with evaluation exposure to the low-level design or implementation. If evidence of developer integration or module testing is used in verifying the expected TSF behaviour, care should be given to confirm that the

testing evidence reflects the current implementation of the TOE. If the subsystem 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.

1013 It should be emphasized 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 of the TSF. This alternative is made available to the developer to minimize the cost (time and/or money) of testing under the circumstances described above; it is not designed to give the evaluator more latitude to demand unwarranted additional information about the TOE, nor to replace testing in general.

5.10.1.3 Verifying the adequacy of tests

1014 Test prerequisites 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 prerequisites for another test. The evaluator must determine that the prerequisites are complete and appropriate in that they will not bias the observed test results towards the expected test results.

1015 The test steps and expected results specify the actions and parameters to be applied to the interfaces 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 functional specification and the high-level design. The tests must verify behaviour documented in these specifications. This means that each TSF behaviour characteristic explicitly described in the functional specification and high-level design should have tests and expected results to verify that behaviour.

1016 Although the entire TSF has to be tested by the developer, exhaustive specification testing of the interfaces is not required. The overall aim of this activity is to determine that the TSF has been sufficiently tested against the behavioural claims in the functional specification and high-level design. The test procedures will provide insight as to how the TSF has been exercised by the developer during testing. The evaluator will use this information when developing additional tests to independently test the TOE.

5.10.2 Evaluation of Coverage (ATE_COV.2)

5.10.2.1 Objectives

1017 The objective of this sub-activity is to determine whether the testing (as documented) is sufficient to establish that the TSF has been systematically tested against the functional specification.

5.10.2.2 Input

a) the ST;

b) the functional specification;

c) the test documentation;

d) the test coverage analysis.

5.10.2.3 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 interfaces in the functional specification.***

4: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.

1018 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.

1019 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the functional specification.

4: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.

1020 Guidance on this work unit can be found in:

a) 5.10.1.1

b) 5.10.1.2

4: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.

1021 Guidance on this work units, as it pertains to the functional specification, can be found in:

a) 5.10.1.3

4: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.

1022 All interfaces 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.

1023 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the functional specification.

5.10.3 Evaluation of Depth (ATE_DPT.1)

5.10.3.1 Objectives

1024 The objective of this sub-activity is to determine whether the developer has tested the TSF against its high-level design.

5.10.3.2 Input

a) the ST;

b) the functional specification;

c) the high-level design;

d) the test documentation;

e) the depth of testing analysis.

5.10.3.3 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 interfaces in the high-level design.***

4:ATE_DPT.1-1 The evaluator ***shall examine*** the depth of testing analysis to determine that the correspondence between the tests in the test documentation and the interfaces in the high-level design is accurate.

1025 A simple cross-table may be sufficient to show test correspondence. The identification of the tests and the interfaces presented in the depth-of coverage analysis has to be unambiguous.

1026 The evaluator is reminded that not all tests in the test documentation must map to an interface in the high-level design.

4:ATE_DPT.1-2 The evaluator ***shall examine*** the test plan to determine that the testing approach for each interface demonstrates the expected behaviour of that interface.

1027 Guidance on this work unit can be found in:

a) 5.10.1.1

b) 5.10.1.2

1028 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.

4:ATE_DPT.1-3 The evaluator ***shall examine*** the test procedures to determine that the test prerequisites, test steps and expected result(s) adequately test each interface.

1029 Guidance on this work units, as it pertains to the high-level design, can be found in:

a) 5.10.1.3

ATE_DPT.1.2C ***The analysis of the depth of testing shall demonstrate that the correspondence between the interfaces in the high-level design and the tests in the test documentation is complete.***

4:ATE_DPT.1-4 The evaluator ***shall examine*** the depth of testing analysis to determine that the correspondence between the interfaces in the high-level design and the tests in the test documentation is complete.

1030 All interfaces that are described in the high-level design have to be present in the depth of testing analysis and mapped to tests in order for completeness to be claimed, although exhaustive specification testing of interfaces is not required. Incomplete depth of testing would be evident if an interface was identified in the high-level design and no tests could be attributed to it.

1031 The evaluator is reminded that this does not imply that all tests in the test documentation must map to interfaces in the high-level design.

5.10.4 Evaluation of Functional tests (ATE_FUN.1)

5.10.4.1 Objectives

1032 The objective of this sub-activity is to determine whether developer correctly performed and documented the tests in the test documentation.

5.10.4.2 Application notes

1033 The extent to which the test documentation is required to cover the TSF is dependent upon the coverage assurance component.

1034 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 security function for which the developer's test results indicate that it may not perform as specified should be tested independently by the evaluator to determine whether or not it does.

5.10.4.3 Input

1035 The evaluation evidence for this sub-activity is:

a) the ST;

b) the test documentation;

5.10.4.4 Action ATE_FUN.1.1E

ATE_FUN.1.1C ***The test documentation shall consist of test plans, expected test results and actual test results.***

4:ATE_FUN.1-1 The evaluator ***shall check*** that the test documentation includes test plans, expected test results and actual test results.

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.***

4:ATE_FUN.1-2 The evaluator ***shall check*** that the test plan identifies the tests to be performed.

1036 The evaluator may wish to employ a sampling strategy when performing this work unit.

4:ATE_FUN.1-3 The evaluator ***shall examine*** the test plan to determine that it describes the scenarios for performing each test.

1037 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.

1038 The evaluator also determines that the test plan provides information about how to execute the test: inputs to be applied, how these inputs are applied, how output is obtained etc. This information should be detailed enough to ensure that the test is reproducible.

1039 The evaluator may wish to employ a sampling strategy when performing this work unit.

4:ATE_FUN.1-4 The evaluator ***shall examine*** the test plan to determine that the TOE test configuration is consistent with the ST.

1040 The TOE referred to in the developer's test plan should have the same unique reference as established in the ST introduction.

1041 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.

1042 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.

1043 The evaluator may wish to employ a sampling strategy when performing this work unit.

4:ATE_FUN.1-5 The evaluator ***shall examine*** the test plans to determine that sufficient instructions are provided for any ordering dependencies.

1044 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 test an audit-related interface before relying on it to produce audit records for testing an access control-related interface. 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.

1045 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.***

4:ATE_FUN.1-6 The evaluator ***shall examine*** the test documentation to determine that all expected tests results are included.

1046 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.

1047 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.***

4:ATE_FUN.1-7 The evaluator ***shall check*** that the expected test results in the test documentation are consistent with the actual test results in the test documentation.

1048 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 synthesize the actual data.

1049 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.).

1050 It should be noted that the description of the process used to reduce or synthesize 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.

1051 The evaluator may wish to employ a sampling strategy when performing this work unit.

4:ATE_FUN.1-8 The evaluator ***shall report*** the developer testing effort, outlining the testing approach, configuration, depth and results.

1052 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.

1053 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;

b) testing approach. An account of the overall developer testing strategy employed;

c) testing results. A description of the overall developer testing results.

1054 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.

5.10.5 Evaluation of Independent testing (ATE_IND.2)

5.10.5.1 Objectives

1055 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.

5.10.5.2 Input

1056 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the user guidance;

e) the administrator guidance;

f) the secure installation, generation, and start-up procedures;

g) the test documentation;

h) the TOE suitable for testing.

5.10.5.3 Action ATE_IND.2.1E

ATE_IND.2.1C ***The TOE shall be suitable for testing.***

4: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.

1057 The TOE referred to in the developer's test plan should have the same unique reference as established in the ST introduction.

1058 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.

1059 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.

1060 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.

4:ATE_IND.2-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state

1061 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 ADO_IGS.1 Installation, generation, and start-up procedures 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, generate and start up the TOE, using the supplied guidance only.

1062 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 ADO_IGS.1-2.

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.***

4: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

1063 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.

5.10.5.4 Action ATE_IND.2.2E

4:ATE_IND.2-4 The evaluator ***shall conduct*** testing using a sample of tests found in the developer test plan and procedures.

1064 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.

1065 Taking into consideration the overall evaluator effort for the entire tests activity, normally 20% of the developer's tests should be performed although this may vary according to the nature of the TOE, and the test evidence supplied.

1066 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.2-4. Additionally, the evaluator may wish to employ a random sampling method to select developer tests to include in the sample.

4:ATE_IND.2-5 The evaluator ***shall check*** that all the actual test results are consistent with the expected test results.

1067 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.

1068 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 even be necessary for the evaluator to increase the sample size, to regain confidence in the developer testing. If the increase in sample size does not satisfy the evaluator's concerns, it may be necessary to repeat the entire set of

developer's tests. Ultimately, 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.

5.10.5.5 Action ATE_IND.2.3E

4:ATE_IND.2-6 The evaluator ***shall devise*** a test subset.

1069 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.

1070 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.

1071 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 coverage analysis, the depth of testing analysis, and the test documentation. 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 rigourously 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.

1072 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) Known public domain weaknesses commonly associated with the type of TOE (e.g. operating system, firewall). Know public domain weaknesses associated with the type of TOE will influence the selection process of the test subset. The evaluator should include those interfaces that are associated with known public domain weaknesses for that type of TOE in the subset (known public domain weaknesses in this context does not refer to vulnerabilities as such but to inadequacies or problem areas that have been experienced with this particular type of TOE).

d) Significance of interfaces. Those interfaces deemed more significant than others should be included in the test subset. An input to this determination could be the number of SFRs mapping to this interface (as determined in Representation correspondence (ADV_RCR)).

e) Complexity of interfaces. Interfaces that require complex implementation may require complex tests that impose onerous requirements on the developer or evaluator, which will 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.

f) Implicit testing. Testing some interfaces may often implicitly test other interfaces, and their inclusion in the subset may maximize 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.

g) 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.

h) 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, the corresponding interfaces should be strong candidates for testing.

1073 This guidance articulates factors to consider during the selection process of an appropriate test subset, but these are by no means exhaustive.

4: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.

1074 With an understanding of the expected behaviour of the TSF, from the ST, the functional specification, and the high-level design, 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).

1075 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.

1076 The evaluator's test documentation should specify the derivation of each test, tracing it back to the relevant interface(s).

4:ATE_IND.2-8 The evaluator ***shall conduct*** testing.

1077 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.

4: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.

1078 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.

1079 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.

4:ATE_IND.2-10 The evaluator ***shall check*** that all actual test results are consistent with the expected test results.

1080 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.

4:ATE_IND.2-11 The evaluator ***shall report*** in the ETR the evaluator testing effort, outlining the testing approach, configuration, depth and results.

1081 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.

1082 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.

1083 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.

5.11 Vulnerability assessment activity

1084 The purpose of the vulnerability assessment activity is to determine the existence and exploitability of flaws or weaknesses in the TOE in the intended environment. This determination is based upon analysis performed by the developer and the evaluator, and is supported by evaluator testing.

5.11.1 Evaluation of Misuse (AVA_MSU.2)

5.11.1.1 Objectives

1085 The objectives of this sub-activity are to determine whether the guidance is misleading, unreasonable or conflicting, whether secure procedures for all modes of operation have been addressed, and whether use of the guidance will facilitate prevention and detection of insecure TOE states.

5.11.1.2 Application notes

1086 The use of the term guidance in this sub-activity refers to the user guidance, the administrator guidance, and the secure installation, generation, and start-up procedures. Installation, generation, and start-up procedures here refers to all procedures the administrator is responsible to perform to progress the TOE from a delivered state to an operational state.

1087 This component includes a requirement for developer analysis that is not present in AVA_MSU.1 Examination of guidance. Validation of this analysis should not be used as a substitute for the evaluator's own examination of the guidance documentation, but should be used to provide evidence that the developer has also explicitly addressed the issue of misuse.

5.11.1.3 Input

1088 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance;

e) the secure installation, generation, and start-up procedures;

f) the misuse analysis of the guidance;

g) TOE suitable for testing.

5.11.1.4 Action AVA_MSU.2.1E

AVA_MSU.2.1C ***The guidance documentation 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.***

4:AVA_MSU.2-1 The evaluator ***shall examine*** the 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.

1089 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.

1090 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.

1091 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 TSP) of that TSFI. The evaluator should also consider the relationships between interfaces, searching for potential conflicts.

AVA_MSU.2.2C ***The guidance documentation shall be complete, clear and reasonable.***

4:AVA_MSU.2-2 The evaluator ***shall examine*** the guidance to determine that it is clear.

1092 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.

4:AVA_MSU.2-3 The evaluator ***shall examine*** the guidance and other evaluation evidence to determine that the guidance is complete and reasonable.

1093 The evaluator should apply familiarity with the TOE gained from performing other evaluation activities to determine that the guidance is complete.

1094 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. The evaluator may, as an aid, prepare an informal mapping between the guidance and these documents. Any omissions in this mapping may indicate incompleteness.

1095 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.

1096 The evaluator should note that results gained during the performance of work units from the Administrator guidance (AGD_ADM) sub-activity will provide useful input to this examination.

AVA_MSU.2.3C ***The guidance documentation shall list all security objectives for the operational environment.***

4:AVA_MSU.2-4 The evaluator ***shall examine*** the guidance to determine that all security objectives for the operational environment are articulated.

1097 The evaluator analyses the security objectives for the operational environment of the TOE as described in the ST and compares them with the guidance to ensure that all security objectives for the operational environment that are relevant to the administrator or user are described appropriately in the guidance.

1098 The guidance should list all external procedural, physical, personnel and connectivity measures, as described in the ST by the security objectives for the operational environment.

AVA_MSU.2.4C ***The analysis documentation shall demonstrate that the guidance documentation is complete.***

4:AVA_MSU.2-5 The evaluator ***shall examine*** the developer's analysis to determine that the developer has taken adequate measures to ensure that the guidance is complete.

1099 The developer analysis may comprise mappings from the ST or the functional specification to the guidance in order to demonstrate that the guidance is complete. Whatever evidence is provided by the developer to demonstrate completeness, the evaluator should assess the developer's

analysis against any deficiencies found during the conduct of work units AVA_MSU.2-1 through AVA_MSU.2-4 and AVA_MSU.2-6.

5.11.1.5 Action AVA_MSU.2.2E

4:AVA_MSU.2-6 The evaluator ***shall perform*** all administrator and user (if applicable) procedures necessary to configure and install the TOE to determine that the TOE can be configured and used securely using only the supplied guidance.

1100 Configuration and installation requires the evaluator to advance the TOE from a deliverable state to the state in which it is operational and enforcing a TSP consistent with the security objectives for the TOE specified in the ST.

1101 The evaluator should follow only the developer's procedures as documented in the user and administrator guidance that is normally supplied to the consumer of the TOE. Any difficulties encountered during such an exercise may be indicative of incomplete, unclear or unreasonable guidance.

1102 Note that work performed to satisfy this work unit may also contribute towards satisfying evaluator action ADO_IGS.1.2E.

4:AVA_MSU.2-7 The evaluator ***shall perform*** other security relevant procedures specified in the guidance to determine that the TOE can be configured and used securely using only supplied guidance.

1103 The evaluator should follow only the developer's procedures as documented in the user and administrator guidance that is normally supplied to the consumer of the TOE.

1104 The evaluator should employ sampling in carrying out this work unit. When choosing a sample the evaluator should consider:

a) the clarity of the guidance - any potential unclear guidance should be included in the sample;

b) guidance that will be used most often - infrequently used guidance should not normally be included in the sample;

c) complexity of the guidance - complex guidance should be included in the sample;

d) severity of error - procedures for which error imparts the greatest severity on security should be included in the sample;

e) the nature of the TOE - the guidance related to the normal or most likely use of the TOE should be included in the sample.

5.11.1.6 Action AVA_MSU.2.3E

4:AVA_MSU.2-8 The evaluator ***shall examine*** the guidance to determine that sufficient guidance is provided for the consumer to effectively administer and use the TSF, and to detect insecure states.

1105 A TOE may use a variety of ways to assist the consumer in effectively using that TOE securely. A TOE may employ functionality (features) to alert the consumer when the TOE is in an insecure state, 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 insecure states.

1106 To arrive at a verdict for this work unit, the evaluator considers the TOE's functionality, its purpose and security objectives for the operational environment. The evaluator should arrive at the conclusion that, if the TOE can transition into an insecure state, there is reasonable expectation that use of the guidance would permit the insecure state to be detected in a timely manner. 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 high-level design of the TSF if a component from High-level design (ADV_HLD) is included).

5.11.1.7 Action AVA_MSU.2.4E

4:AVA_MSU.2-9 The evaluator ***shall examine*** the developer's analysis of the guidance to determine that guidance is provided for secure operation in all modes of operation of the TOE.

1107 The results of evaluation action AVA_MSU.2.1E should provide a basis with which to evaluate the developer's analysis. Having evaluated the potential for misuse of the guidance, the evaluator should be able to determine that the developer's misuse analysis meets the objectives of this sub-activity.

5.11.2 Evaluation of Vulnerability analysis (AVA_VLA.2)

5.11.2.1 Objectives

1108 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.

5.11.2.2 Application notes

1109 An exploitable vulnerability is a weakness or flaw in the TOE. An attack is a way of exploiting weaknesses or flaws.

1110 When considering a vulnerability in relation to the TOE, the vulnerability is prefixed with potential, residual or exploitable. However, those in the public domain, and therefore not in relation to the TOE are generally simply referred to as vulnerabilities, if there is an environment in which they can be exploited.

1111 The use of the term guidance in this sub-activity refers to the user guidance, the administrator guidance, and the secure installation, generation, and start-up procedures.

1112 The consideration of exploitable vulnerabilities will be determined by the security objectives for the operational envrionment and SFRs in the ST. For example, if security objectives for the operational environment state the TOE is to be physically protected, any attacks requiring physical access to the TOE should not be considered.

1113 Potential vulnerabilities may be in the public domain, 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 in the public domain, it can be easily exploited.

5.11.2.3 Input

1114 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design;

d) the low-level design;

e) the implementation subset selected;

f) the guidance documentation;

g) the vulnerability analysis;

h) the TOE suitable for testing.

1115 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.

1116 Other input for this sub-activity is:

a) current information regarding public domain vulnerabilities and attacks (e.g. from an overseer).

5.11.2.4 Action AVA_VLA.2.1E

AVA_VLA.2.1C ***The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.***

AVA_VLA.2.2C ***The vulnerability analysis documentation shall describe the disposition of identified potential vulnerabilities.***

AVA_VLA.2.3C ***The vulnerability analysis documentation shall show, for all identified potential vulnerabilities, that the vulnerability cannot be exploited in the operational environment for the TOE.***

4:AVA_VLA.2-1 The evaluator ***shall examine*** the developer's vulnerability analysis to determine that the search for potential vulnerabilities has considered all relevant information.

1117 The developer's vulnerability analysis should cover the developer's search for potential vulnerabilities in at least all evaluation deliverables and public domain information sources (the developer reports the sources used in the vulnerability analysis).

1118 Information in the public domain is highly dynamic. Therefore, it is possible that new vulnerabilities are reported in the public domain between the time the developer performs the vulnerability analysis and the time that the evaluation is completed. The point at which monitoring of the public domain information ceases is an evaluation authority issue; therefore guidance should be sought from the evaluation authority.

4:AVA_VLA.2-2 The evaluator ***shall examine*** the developer's vulnerability analysis to determine that each identified potential vulnerability is described and that a rationale is given for why it is not exploitable in the operational environment for the TOE.

1119 The developer is expected to search for potential vulnerabilities, based on knowledge of the TOE as documented in the inputs provided, and of public domain information sources. The developer shows that potential vulnerabilities are not exploitable in the operational environment.

1120 The evaluator needs to be concerned with two aspects of the developer's analysis:

a) whether the developer's analysis has considered all evaluation deliverables;

b) whether appropriate measures are in place to prevent the exploitation of potential vulnerabilities in the operational environment.

1121 A potential vulnerability is termed non-exploitable if one or more of the following conditions exist:

a) measures in the operational environment, either IT or non-IT, prevent exploitation of the 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;

b) the vulnerability is exploitable but only by attackers possessing moderate or high attack potential. For example, a potential vulnerability of a distributed TOE to session hijack attacks might require bespoke equipment and critical knowledge of the TOE which indicates an attack potential beyond basic. However, such vulnerabilities are reported in the ETR as residual vulnerabilities;

c) it does not prevent the TOE from meeting its SFRs, and hence the TSP, in the operational environment. For instance, a firewall whose ST includes no availability requirements (Availability of exported TSF data (FPT_ITA) or FRU: Resource utilisation components) and makes no availability policy claim, but is vulnerable to TCP SYN attacks (an attack on a common Internet protocol that renders hosts incapable of servicing connection requests) should not fail this evaluator action on the basis of this vulnerability alone.

1122 If a vulnerability can be exploited in the TOE in its operational environment, but it not apparent that the level of attack potential required is basic, the guidance on determining the necessary attack potential can be found in Chapter 8.13.

4:AVA_VLA.2-3 The evaluator ***shall examine*** the developer's vulnerability analysis to determine that it is consistent with the ST and the guidance.

1123 The developer's vulnerability analysis may address a potential vulnerability by suggesting specific configurations or settings for the TOE. If such operating constraints are deemed to be effective and consistent with the ST, then all such configurations/settings should be adequately described in the guidance so that they may be employed by the consumer.

5.11.2.5 Action AVA_VLA.2.2E

4:AVA_VLA.2-4 The evaluator ***shall devise*** penetration tests to verify on the developer vulnerability analysis .

1124 The evaluator prepares for penetration testing:

a) as necessary to attempt to disprove the developer's analysis in cases where the developer's rationale for why a vulnerability is unexploitable is suspect in the opinion of the evaluator;

b) as necessary to determine the susceptibility of the TOE, in its operational environment, to additional potential vulnerabilities to be considered by the evaluator. These additional potential vulnerabilities may be sourced from current information (e.g. from the overseer) regarding public domain vulnerabilities that may not have been considered by the developer, and may also have been identified as potential vulnerabilities as a result of performing other evaluation activities.

1125 The evaluator is not expected to test for potential vulnerabilities (including those in the public domain) beyond those for which a basic attack potential is required to effect an attack. In some cases, however, it will be necessary to carry out a test before the attack potential required can be determined. Where, as a result of evaluation expertise, the evaluator discovers a vulnerability that is beyond basic attack potential, this is reported in the ETR as a residual vulnerability.

1126 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 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.

1127 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.

4:AVA_VLA.2-5 The evaluator ***shall examine*** all inputs to this sub-activity to determine potential vulnerabilities not already addressed by the developer's vulnerability analysis.

1128 A focused 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 a 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.

1129 If a vulnerability can be exploited in the TOE, but it not apparent that the level of attack potential required is basic, the guidance on determining the necessary attack potential can be found in Chapter 8.13.

1130 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.

1131 Potential vulnerabilities hypothesised exploitable by an attacker possessing a basic attack potential, that do not result in a violation of the security objectives specified in the ST, 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.

1132 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.

1133 Subject to the SFRs the TOE is to meet in the operational environment, the evaluator's independent vulnerability analysis should consider generic vulnerabilities under each of the following headings:

a) generic vulnerabilities relevant for the type of TOE being evaluated, as may be supplied by the overseer;

b) bypassing;

c) tampering;

d) direct attacks;

e) misuse.

1134 Items b) - e) are now explained in greater detail.

5.11.2.5.1 Bypassing

1135 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.

1136 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) invocating 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. The use of undocumented interfaces (which may be insecure) also falls into this category (these include undocumented support and help facilities).

e) Using implementation detail introduced in lower representations again includes the use of covert channels in which an attacker takes advantage of additional functions, resources or attributes that are introduced to the TOE as a consequence of the refinement process (e.g. use of a lock variable as a covert channel). Additional functionality may also include test harness code contained in software modules.

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 on 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 disc 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.

5.11.2.5.2 Tampering

1137 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.

1138 Each of the following should be considered (where relevant) in the evaluator's independent vulnerability analysis.

a) Attacks based on accessing data on 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 decive 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);

3) an end-of-file marker (e.g. CTRLZ or CTRLD) 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 discs;

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 CTRLC, CTRLY, 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.

5.11.2.5.3 Direct attacks

1139 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.

1140 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.

1141 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.

1142 For example, the maximum theoretical password space (i.e. all printable ASCII characters) would not be worst case because it is human behaviour to use natural language passwords, effectively reducing the password space and associated strength. However, such an assumption could be appropriate if the SFRs in the ST included FIA_SOS.1 Verification of secrets or FIA_SOS.2 TSF Generation of secrets to minimise the use of natural language passwords.

1143 Probabilistic or permutational mechanisms should be identified during examination of evaluation evidence required as input to this sub-activity (security target, functional specification, high-level design, low-level design and implementation representation subset) and any other TOE (e.g. guidance) documentation may identify additional probabilistic or permutational mechanisms.

1144 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.

1145 Direct attacks reliant upon a weakness in a cryptographic algorithm should not be considered under Vulnerability analysis (AVA_VLA), as this is outside the scope of the CC. Correctness of the implementation of the cryptographic algorithm is considered during the ADV and ATE: Tests activities.

5.11.2.5.4 Misuse

1146 Misuse includes the identification of any penetration tests necessary to confirm or disprove the misuse analysis. Issues to be considered include:

a) behaviour of the TOE when start-up, closedown 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.

4:AVA_VLA.2-6 The evaluator ***shall devise*** penetration tests, based on the independent vulnerability analysis.

1147 The evaluator prepares for penetration testing based on the prioritised list of potential vulnerabilities hypothesised in work unit AVA_VLA.2-5.

1148 The evaluator is not expected to test 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 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.

1149 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 interfaces 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.

1150 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.

4:AVA_VLA.2-7 The evaluator ***shall produce*** penetration test documentation for the tests 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.

1151 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.

4:AVA_VLA.2-8 The evaluator ***shall conduct*** penetration testing.

1152 The evaluator uses the penetration test documentation resulting from work unit AVA_VLA.2-4 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 learned 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.

1153 The evaluator uses the penetration test documentation resulting from work unit AVA_VLA.2-10 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 new tests as a result of information learned 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.

1154 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.

4:AVA_VLA.2-9 The evaluator ***shall record*** the actual results of the penetration tests.

1155 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.

4:AVA_VLA.2-10 The evaluator ***shall report*** in the ETR the evaluator penetration testing efforts, outlining the testing approach, configuration, depth and results.

1156 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 overseers 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.

1157 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 that were the focus of the penetration testing;

c) verdict for the sub-activity. The overall judgement on the results of penetration testing.

1158 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.

4:AVA_VLA.2-11 The evaluator ***shall examine*** the results of all penetration testing and the conclusions of all vulnerability analysis to determine that the TOE, in its operational environment, is resistant to an attacker possessing a basic attack potential.

1159 If the results reveal that the TOE, in its operational environment, has vulnerabilities exploitable by an attacker possessing less than a moderate attack potential, then this evaluator action fails.

4:AVA_VLA.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 SFRs not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual);

e) identification of evaluation party (e.g. developer, evaluator) who identified it.

6 Flaw remediation sub-activities

6.1 Evaluation of flaw remediation (ALC_FLR.1)

6.1.1 Objectives

1160 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.

6.1.2 Input

1161 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation.

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.

1162 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 timeframe, from initial detection through ascertaining the flaw is a security flaw, to resolution of the security flaw.

1163 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.

1164 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.

1165 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.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.

1166 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.

1167 *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).

1168 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.

1169 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.

1170 TOE users may be provided 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.

1171 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.

6.2 Evaluation of flaw remediation (ALC_FLR.2)

6.2.1 Objectives

1172 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.

1173 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

6.2.2 Input

1174 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation;

b) flaw remediation guidance documentation.

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.

1175 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 timeframe, from initial detection through ascertaining the flaw is a security flaw, to resolution of the security flaw.

1176 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.

1177 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.

1178 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.

1179 *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).

1180 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.

1181 *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.

1182 TOE users may be provided 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.

1183 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.

1184 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.

1185 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 corrected and the correction 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.

1186 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.

1187 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 corrective actions for each security flaw.

1188 The procedures describe the process that is taken from the point at which a security flaw is resolved to the point at which the corrective action is 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 ADO_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 expected to be distributed by bonded courier. Updates unrelated to flaw remediation would follow the procedures set forth in the documentation meeting the Delivery (ADO_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.

1189 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.

1190 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.

1191 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.

6.3 Evaluation of flaw remediation (ALC_FLR.3)

6.3.1 Objectives

1192 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.

1193 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.

6.3.2 Input

1194 The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation;

b) flaw remediation guidance documentation.

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.

1195 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 timeframe, from initial detection through ascertaining the flaw is a security flaw, to resolution of the security flaw.

1196 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.

1197 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.

1198 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.

1199 *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).

1200 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.

1201 *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.

1202 TOE users may be provided 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.

1203 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.

1204 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.

1205 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.

1206 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 procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.***

ALC_FLR.3-7 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 corrected.

1207 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.

1208 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.3-8 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 corrective actions for each security flaw.

1209 The procedures describe the process that is taken from the point at which a security flaw is resolved to the point at which the corrective action is 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 Delivery (ADO_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 expected to be distributed by bonded courier. Updates unrelated to flaw remediation would follow the procedures set forth in the documentation meeting the Delivery (ADO_DEL) requirements.

ALC_FLR.3.7C ***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.3-9 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.

1210 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.

1211 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.3.8C ***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.3-10 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.

1212 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.3.9C ***The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.***

ALC_FLR.3-11 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.

1213 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.

1214 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.3-12 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.

1215 *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.

1216 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.3.10C ***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.

1217 *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.

1218 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.

1219 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.

1220 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.

1221 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.

A Glossary

(normative)

1222 This annex presents abbreviations, acronyms and vocabulary used by the CEM and does not include those already presented in the CC. This annex also presents the references used in the CEM.

A.1 Abbreviations and acronyms

1223 CEM Common Methodology for Information Technology Security Evaluation

1224 ETR Evaluation Technical Report

1225 OR Observation Report

A.2 Vocabulary

1226 Terms which are presented in bold-faced type are themselves defined in this section.

1227 Check :

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

1228 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.

1229 Evaluation Evidence :

> a tangible **evaluation deliverable**.

1230 Evaluation Technical Report :

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

1231 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.

1232 Interpretation :

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

1233 Methodology :

the system of principles, procedures and processes applied to IT security evaluations.

1234 Observation Report :

a report written by the evaluator requesting a clarification or identifying a problem during the evaluation.

1235 Overall Verdict :

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

1236 Oversight Verdict :

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

1237 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.

1238 Report :

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

1239 Scheme :

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

1240 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.

1241 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**.

A.3 References

CC Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999.

COD Concise Oxford Dictionary, Oxford University Press, Ninth edition, 1995.

IEEE IEEE Standard Glossary of Software Engineering Terminology, ANSI/IEEE STD 729-1983

B General evaluation guidance (normative)

B.1 Objectives

1242 The objective of this chapter is to cover the basic evaluation techniques used to provide technical evidence of evaluation results. The use of such techniques helps in achieving objectivity, repeatability and reproducibility of the work performed by the evaluator.

B.2 Sampling

1243 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.

1244 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.

1245 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.

1246 The CC identifies the following evaluator action elements where sampling is explicitly acceptable:

a) ADV_RCR.3.2E: “The evaluator shall determine the accuracy of the proofs of correspondence by selectively verifying the formal analysis.”

b) Independent testing (ATE_IND).*.2E: “The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified”.

c) ATE_IND.2.3E: “The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.”

d) Covert channel analysis (AVA_CCA).*.3E: “The evaluator shall selectively validate the covert channel analysis through testing.”

e) AVA_MSU.2.2E and AVA_MSU.3.2E: “The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.”

1247 In addition ADV_IMP.1.1D requires that the developer provide the implementation representation for a subset of the TSF only. The sample of the subset should be selected in agreement with the evaluator. Provision of a sample of the implementation representation allows the evaluator to assess the presentation of the implementation representation itself and to sample the traceability evidence to gain assurance in the correspondence between the low-level design and the implementation representation.

1248 In addition to the sampling that the CC accepts, the CEM identifies the following actions where sampling is acceptable:

a) Action CM capabilities (ACM_CAP).*.1E: “The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.”

Here sampling is accepted for the content and presentation of evidence elements CM capabilities (ACM_CAP).*.8C and CM capabilities (ACM_CAP).*.9C for EAL3 and EAL4.

b) Action ATE_FUN.1.1E: “The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.”

Here sampling is accepted for the content and presentation of evidence element ATE_FUN.1.2C, ATE_FUN.1.3C, and ATE_FUN.1.4C for EAL2, EAL3, and EAL4.

c) Action ALC_DVS.1.1E: “The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.”

Here sampling is accepted for the content and presentation of evidence element ALC_DVS.1.2C for EAL3 and EAL4.

1249 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.

1250 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.

1251 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 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 here will differ from that taken when sampling a developer's test results. 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.

1252 The following principles should be followed whenever sampling is performed:

a) 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), but a minimum size of 20% should be adopted as a norm for sampling material related to the TOE implementation. Where sampling relates to gaining evidence that a process (e.g. visitor control or design review) is being followed, a percentage figure is not appropriate. The evaluator should sample sufficient information to gain reasonable confidence that the process is being followed, and justify the sample size.

b) The sample should be representative of all aspects relevant to the areas that are sampled. In particular, a selection should cover a variety of components, security functions, developer and operational sites (if more than one is involved) and hardware platform types (if more than one is involved).

c) 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.

d) 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.

1253 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.

B.3 Dependencies

1254 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.

B.3.1 Dependencies between activities

1255 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.

B.3.2 Dependencies between sub-activities

1256 Dependencies identified between components in CC Part 3 have to be considered by the evaluator. An example for this kind of dependency is AVA_VLA.1 Developer vulnerability analysis. This component claims dependencies on ADV_FSP.1 Informal functional specification, ADV_HLD.1 Descriptive high-level design, AGD_ADM.1 Administrator guidance and AGD_USR.1 User guidance.

1257 A sub-activity can be assigned a pass verdict normally only if all those sub-activities are successfully completed on which it has a dependency. For example, a pass verdict on AVA_VLA.1 Developer vulnerability analysis can normally only be assigned if the sub-activities related to ADV_FSP.1 Informal functional specification, ADV_HLD.1 Descriptive high-level design, AGD_ADM.1 Administrator guidance and AGD_USR.1 User guidance are assigned a pass verdict too.

1258 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.

1259 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.

B.3.3 Dependencies between actions

1260 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.

B.4 Site Visits

1261 This section provides general guidance on site visits. Specific and detailed information is given in work units for those activities where site visits are performed:

a) CM automation (ACM_AUT);

b) CM capabilities (ACM_CAP).n (with n>2);

c) Delivery (ADO_DEL);

d) Development security (ALC_DVS).

1262 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.

1263 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;

c) to observe the application of security measures during development.

1264 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.

1265 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.

1266 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 practice 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.

1267 To prepare for the site visit a checklist, based on the evaluation evidence provided should be generated by the evaluator. The results of the site visit should be recorded.

1268 Site visits may not be deemed necessary if e.g. the development site has recently been visited for another TOE evaluation or particular ISO 9000 procedures were confirmed as being followed. 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 overseer.

B.5 TOE Boundary

1269 The identity of what is evaluated will appear in the ETR, on the certificate, in the ST, and on the list of evaluated products. Although products are typically bought and sold, evaluations are concerned with TOEs. In cases where the developer of the product is also the developer of the evaluation evidence (i.e. the sponsor), this distinction is unnecessary. But because these roles may be filled by different parties, the following were agreed as the basis of definitions used in the CEM, along with their interrelationships and effects upon evaluations and certification.

B.5.1 Product and system

1270 The product is the collection of hardware and/or software that is available for use. Some purveyors might bundle a collection of products (e.g. a wordprocessor, spreadsheet, and graphics application) into yet another product (e.g. an office automation system). But, provided that it is available for use, either by the public, by other manufacturers, or by limited customers, the resulting collection is considered to be a product.

1271 A system consists of one or more products in a known operational environment. The main difference between a product evaluation and a

system evaluation is that, for a system evaluation, the evaluator takes into account the actual environment instead of theorising a hypothetical one, as done for a product evaluation.

B.5.2 TOE

1272 The TOE is the entity that is evaluated as defined by the ST. While there are cases where a TOE makes up the entire product, this need not be the case. The TOE may be a product, a part of a product, a set of products, a unique technology never to be made into a product, or combinations of all of these, in a specific configuration or set of configurations. This specific configuration or set of configurations is called the evaluated configuration. The ST clearly describes the relation between the TOE and any associated products.

1273 This evaluated configuration is identified in sufficient detail to differentiate hardware included in the evaluated configuration from hardware that is not included in the evaluated configuration, though it might be available as part of the product upon which the TOE is based. This identification makes it apparent to potential customers what product must be purchased, and what configuration options must be used, in order for the TOE to run securely.

B.5.3 TSF

1274 The TSF is the collection of those parts of the TOE that must be relied upon to enforce the security of the TOE as defined by the SR. There may be parts within the TOE that contribute nothing to the security of the TOE as defined by the ST; consequently, such parts would not be part of the TSF.

1275 The hardware portions of the TSF are described at a level of detail commensurate with the assurance requirements related to the relevant development documentation (functional specification, high-level design, low-level design) and the testing documentation. The level of hardware identification is determined by the impact that the hardware features have upon the security functions and assurances being claimed.

B.5.4 Evaluation

1276 An implicit assumption for all evaluations is that the TOE is (by definition) the product or system in its evaluated configuration; this assumption need not be explicitly included in the list of assumptions for the evaluation. The TOE undergoes the scrutiny of the evaluation: analysis is performed only within the evaluated configuration, testing is performed upon this evaluated configuration, exploitable vulnerabilities are identified in this evaluated configuration, and assumptions are relevant only in the evaluated configuration. The ease with which the TOE can exit this configuration is important, and must be considered where Misuse (AVA_MSU) is called up. This will look at the robustness of the TOE configuration, and the impact of any accidental or intentional deviations from it that may occur without detection.

1277 The following example provides three TOEs, all of which are based upon the same virtual private networking (VPN) firewall product, but which yield different evaluation results because of the differences in the STs.

1278 **1) A VPN-firewall which is configured in such a way that the VPN functionality is turned off. All threats in the ST are concerned with access to the safe network from the unsafe network.**

1279 The TOE is the VPN-firewall configured in such a way that the VPN functionality is turned off. If the administrator were to configure the firewall such that some or all VPN functions were enabled, the product would not be in an evaluated configuration; it would therefore be considered to be unevaluated, and so nothing could be stated about its security.

1280 **2) A VPN-firewall, where all threats in the ST are concerned with access to the safe network from the unsafe network.**

1281 The TOE is the entire VPN-firewall. The VPN functions are part of the TOE, so one of the things to be determined during the evaluation would be whether there are means to gain access to the safe network from the unsafe network through the VPN functions.

1282 **3) A VPN-firewall, where all threats in the ST are concerned with either access to the safe network from the unsafe network or confidentiality of traffic on the unsafe network.**

1283 The TOE is the entire VPN-firewall. The VPN functions are part of the TOE, so one of the things to be determined during the evaluation would be whether the VPN functions permit the realisation of any of the threats described in the ST.

B.5.5 Certification

1284 From the earlier paragraphs, it is clear that evaluating the same product with different STs leads to different TOEs with different TSFs. Consequently, the Certificates, ETR, the STs, and the entries in the Evaluated Products List will have to differ among the evaluations to be of any use to potential customers.

1285 Note that, for the above example of three different firewall evaluations, the apparent differences between these Certificates would be subtle, as the three VPN-firewalls would all lead to certificates identifying the TOE as:

1286 *The XYZ Firewall product, as described in the Evaluated Configuration identified in Security Target #ABC.*

1287 with a different identifier for each ST ABC.

1288 Therefore, the evaluator has to ensure that the ST adequately describes the TOE in terms of what functionality is within the scope of the evaluation. A clear explanation is vital because prospective customers of evaluated products will consult the STs of the products that they are considering to buy

in order to determine which security functionality of those products have been evaluated.

B.6 Impact of FTP on the Assurance Families

1289 The inclusion/exclusion of the FPT self-protection requirements from the PP/ST will affect the following requirements:

B.6.1 ADV

1290 Where the threat of tampering or bypass does not exist, the evaluation will focus on correct operation of the TSF. This will include consideration of all parts of the TOE that contribute directly or indirectly to the enforcement of the TSP. Parts that fall into neither of these categories need not be examined (the presence of errors in these parts that can interfere with the correct operation of the TSF will be established through testing of the TSF).

1291 Where self-protection functions have been claimed, the description of their implementation will identify the protection mechanisms, from which a determination of the TSF boundaries can be made. Identification of the TSF boundaries and interfaces, together with a determination of the efficacy of the TSF protection mechanisms claimed, will allow the evaluation to be limited in scope. This limitation will exclude parts outside the TSF, since these cannot interfere with correct TSF operation. In many cases, the TSF boundary will include some parts that do not contribute to the enforcement of the TSP, and these partss will need to be examined during the evaluation. Those parts that can be determined not to fall within the TSF need not be examined by the evaluator.

B.6.2 ATE_IND

1292 The application notes for Independent testing (ATE_IND) call for testing of obvious public domain weaknesses that may be applicable to the TOE. Such weaknesses that are based on the intent to tamper or bypass the TOE need only be considered where such a threat has been identified.

B.7 Scheme Responsibilities

1293 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.)

1294 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 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;