Australian Government
Department of Defence

Defence Signals Directorate

Australasian Information Security Evaluation Program

Functional Specification (ADV_FSP.4) - CC V2.2

Common Evaluation Methodology

15 February 2006

Version 1.1

Amendment Record

Version	Date	Description
1.0	20th June 2005	Released.
1.1	15 February 2006	Releasable to AISEFs.

Table of Contents

List of Figures

1 Formal Functional Specification (ADV_FSP.4)

1.1 Objectives

1 The objective of this sub-activity is to determine whether the developer has provided an adequate description of all security functions of the TOE and whether the security functions provided by the TOE are sufficient to satisfy the security functional requirements of the ST.

1.2 Input

2 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the user guidance;

d) the administrator guidance.

1.3 Evaluator Actions

1.3.1 ADV_FSP.4.1E

ADV_FSP.4.1C The functional specification shall describe the TSF and its external interfaces using a ***formal*** style, supported by informal, explanatory text where appropriate.

ADV_FSP.4-1 The evaluator shall examine the functional specification to determine that it describes the TSF and its external interfaces using a formal style.

3 TSF and TSFI are to be specified using an appropriate formal notation agreed by the Scheme.

ADV_FSP.4-2 The evaluators ***shall examine*** the functional specification to determine that all formal arguments are correct and consistent with the formal notation syntactic and semantic rules.

4 Each TSF and TSFI formal argument is to be assessed to determine that it is valid and consistent with the syntactic and semantic rules of the formal notation used.

ADV_FSP.4-3 The evaluator ***shall examine*** the functional specification to determine that it contains all necessary informal explanatory text.

5 Supporting narrative descriptions are necessary for those portions of the functional specification that are difficult to understand.

ADV_FSP.4.2C The functional specification shall be internally consistent.

ADV_FSP.4-4 The evaluator ***shall examine*** the functional specification to determine that it is internally consistent.

6 The evaluator validates the functional specification by ensuring that the descriptions of the interfaces making up the TSFI are consistent with the descriptions of the functions of the TSF.

7 Evaluators should check that informal prose does not enumerate any security relevant facts that are not covered by the formal description.

ADV_FSP.4.3C 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.

ADV_FSP.4-5 The evaluator ***shall examine*** the functional specification to determine that it identifies all of the external TOE security function interfaces.

8 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 1 shows a TOE with TSF (cross-hatched) portions and non-TSF (empty) portions. This TOE has three external interfaces:

a) interface c is a direct interface to the TSF;

b) interface b is an indirect interface to the TSF; and

c) interface a is an interface to non-TSF portions of the TOE.

9 Therefore, interfaces b and c make up the TFSI.

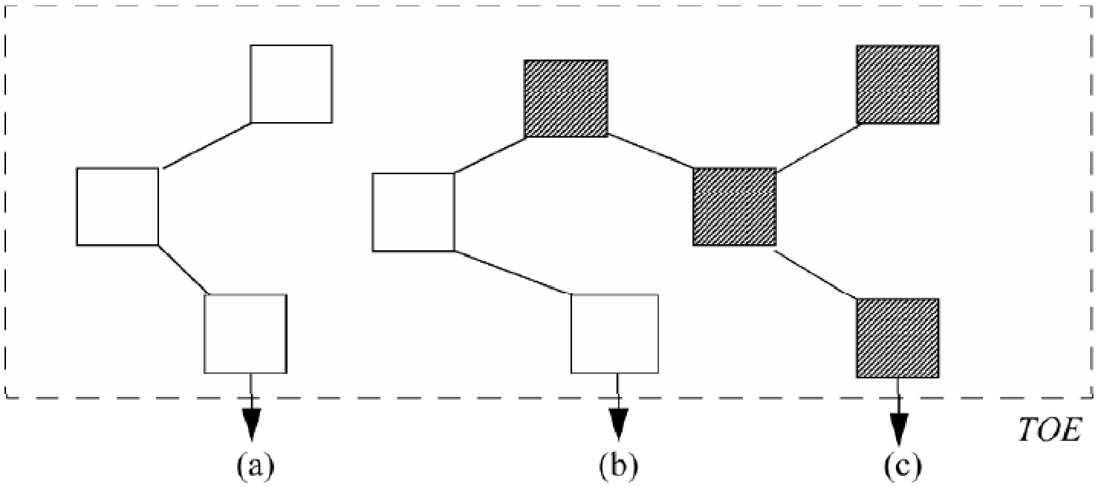

Figure 1 – TSF Interfaces

10 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.

ADV_FSP.4-6 The evaluator ***shall examine*** the functional specification to determine that it describes all of the external TOE security function interfaces.

11 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.

12 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 1) 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 1). 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.

13 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 affecting the TSP (i.e. such programs are interfaces of type a, rather than b in Figure 1) 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.

ADV_FSP.4-7 The evaluator ***shall examine*** the presentation of the TSFI to determine that it adequately and correctly describes the complete behaviour of the TOE at each external interface describing effects, exceptions and error messages.

14 In order to assess the adequacy and correctness of an interface's presentation, the evaluator uses the functional specification, the TOE summary specification of the ST, 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.

15 Verification of the above is done by reviewing the functional specification and the TOE summary specification of the ST, 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 under evaluation; the evaluator would also ensure that the processor instruction set is described.

16 This review might be iterative, such that the evaluator would not discover the functional specification to be incomplete until the 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.4.4C	The functional specification shall completely represent the TSF.

ADV_FSP.4-8 The evaluator ***shall examine*** the functional specification to determine that the TSF is fully represented.

17 In order to assess the completeness of the TSF representation, the evaluator consults the TOE summary specification of the ST, the user guidance, and the administrator guidance. None of these should describe security functions that are absent from the TSF presentation of the functional specification.

ADV_FSP.4.5C The functional specification shall include rationale that the TSF is completely represented.

ADV_FSP.4-9 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.

18 The evaluator determines that there is a convincing argument that there are no interfaces of 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 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.

1.3.2 ADV_FSP.4.2E

ADV_FSP.4.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_FSP.4-10 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the TOE security functional requirements.

19 To ensure that all ST security functional requirements are covered by the functional specification, the evaluator may construct a map between the TOE summary specification 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 security functional requirements are mapped onto applicable TSFI presentations in the functional specification.

ADV_FSP.4-11 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the TOE security functional requirements.

20 For each interface to a security function with specific characteristics, the detailed information in the functional specification must be exactly as it is specified in the ST. For example, if the ST contains user authentication requirements that the password length must be eight characters, the TOE must have eight-character passwords; if the functional specification describes six-character fixed length passwords, the functional specification would not be an accurate instantiation of the requirements.

21 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 security requirements; 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 that this is appropriate (because, perhaps, access mediation is performed on READs and WRITEs, rather than on OPENs).