Australian Government
Department of Defence

Defence Signals Directorate

Australasian Information Security Evaluation Program

High Level Design (ADV_HLD.5) - CC V2.2

Common Evaluation Methodology

15 February 2006

Version 1.2

Amendment Record

Version	Date	Description
1.0	08/7/2005	Released.
1.1	15/8/2005	Released to AISEFs.
1.2	15/2/2006	Added CC V2.2 to title.

Table of Contents

1 FORMAL HIGH LEVEL DESIGN (ADV_HLD.5) 4

1.1 OBJECTIVES 4
1.2 INPUT 4
1.3 EVALUATOR ACTIONS 4
1.3.1 ADV_HLD.5.1E 4
1.3.2 ADV_HLD.5.2E 11

1 Formal High Level Design (ADV_HLD.5)

1.1 Objectives

1 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.

1.2 Input

2 The evaluation evidence for this sub-activity is:

a) the ST;

b) the functional specification;

c) the high-level design.

1.3 Evaluator Actions

1.3.1 ADV_HLD.5.1E

ADV_HLD.5.1C	The presentation of the high-level design shall be ***formal***.

ADV_HLD.5-1 The evaluator shall examine the high-level design to determine that it describes the subsystems using a formal style.

3 Each TSF Subsystem is to be specified using an appropriate formal notation agreed by the Scheme.

ADV_HLD.5-2 The evaluators ***shall examine*** the high-level design to determine that all formal arguments are correct and consistent with the formal notation syntactic and semantic rules.

4 Each TSF subsystem 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_HLD.5-3 The evaluator ***shall examine*** the high-level design to determine that it contains all necessary informal explanatory text.

5 Supporting narrative descriptions are necessary for those portions of the high-level design that are difficult to understand only from formal description.

ADV_HLD.5.2C The high-level design shall be internally consistent.

ADV_HLD.5-4 The evaluator ***shall examine*** the presentation of the high-level design to determine that it is internally consistent.

6 For guidance on consistency analysis see B.3.

7 The evaluator validates the subsystem interface specifications by ensuring that the interface specifications are consistent with the description of the purpose of the subsystem.

ADV_HLD.5.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.5-5 The evaluator ***shall examine*** the high-level design to determine that the TSF is described in terms of subsystems.

8 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.

9 The primary purpose for examining the high-level design is to aid the evaluator's understanding of the TOE. The developer's choice of subsystem definition, and of the grouping of TSFs within each subsystem, are an important aspect of making the high-level design useful in understanding the TOE's intended operation. As part of this work unit, the evaluator should make an assessment as to the appropriateness of the number of subsystems presented by the developer, and also of the choice of grouping of functions within subsystems. 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.

10 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”.

11 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.5-12.

ADV_HLD.5.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.5-6 The evaluator ***shall examine*** the high-level design to determine that it describes the security functionality of each subsystem.

12 The security functional behaviour 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 through its functions and the effects the subsystem may have on the security state of the TOE (e.g. changes in subjects, objects, security databases).

ADV_HLD.5.5C 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.

ADV_HLD.5-7 The evaluator ***shall check*** the high-level design to determine that it identifies all hardware, firmware, and software required by the TSF.

13 If the ST contains no security requirements for the IT environment, this work unit is not applicable and is therefore considered to be satisfied.

14 If the ST contains the optional statement of security requirements for the IT environment, the evaluator compares the list of hardware, firmware, or software required by the TSF as stated in the high-level design to the statement of security requirements for the IT environment to determine that they agree. The information in the ST characterises the underlying abstract machine on which the TOE will execute.

15 If the high-level design includes security requirements for the IT environment that are not included in the ST, or if they differ from those included in the ST, this inconsistency is assessed by the evaluator under Action ADV_HLD.5.2E.

ADV_HLD.5-8 The evaluator ***shall examine*** the high-level design to determine that it includes a presentation of the functions provided by the supporting protection mechanisms implemented in the underlying hardware, firmware, or software.

16 If the ST contains no security requirements for the IT environment, this work unit is not applicable and is therefore considered to be satisfied.

17 The presentation of the functions provided by the underlying abstract machine on which the TOE executes need not be at the same level of detail as the presentation of functions that are part of the TSF. The presentation should explain how the TOE uses the functions provided in the hardware, firmware, or software that implement the security requirements for the IT environment that the TOE is dependent upon to support the TOE security objectives.

18 The statement of security requirements for the IT environment may be abstract, particularly if it is intended to be capable of being satisfied by a variety of different combinations of hardware, firmware, or software. As part of the Tests activity, where the evaluator is provided with at least one instance of an underlying machine that is claimed to satisfy the security requirements for the IT environment, the evaluator can determine whether it provides the necessary security functions for the TOE. This determination by the evaluator does not require testing or analysis of the underlying machine; it is only a determination that the functions expected to be provided by it actually exist.

ADV_HLD.5.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.5-9 The evaluator ***shall check*** that the high-level design identifies the interfaces to the TSF subsystems.

19 The high-level design includes, for each subsystem, the name of each of its entry points.

ADV_HLD.5.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.5-10 The evaluator ***shall check*** that the high-level design identifies which of the interfaces to the subsystems of the TSF are externally visible.

20 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.5.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing complete details of all effects, exceptions and error messages.

ADV_HLD.5-11 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 complete details of all effects, exceptions and error messages.

21 The high-level design must describe the purpose and intended method of use of all subsystem interfaces. In doing so, each interface shall be described in complete detail.

22 Detailed descriptions shall include all input and output parameters, the effects of the interface, and all exceptions and error messages it produces.

23 In the case of external interfaces, the required description may be included in the high-level design and can be referenced in the high-level design without replication.

24 The evaluator should seek to understand the purpose and behaviour of each subsystem, and the nature of the interactions between subsystems to establish confidence that the TOE design is sound.

ADV_HLD.5.9C The high-level design shall describe the separation of the TOE into TSP enforcing and other subsystems.

ADV_HLD.5-12 The evaluator ***shall check*** that the high-level design describes the separation of the TOE into TSP-enforcing and other subsystems.

25 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 functions that directly enforce the TSP, and also those functions 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.

26 As explained under work unit ADV_HLD.5-5, the developer's choice of subsystem definition, and of the grouping of TSFs within each subsystem, are important aspects of making the high-level design useful in understanding the TOE's intended operation. However, the choice of grouping of TSFs within 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.

ADV_HLD.5.10C The high-level design shall justify that the identified means of achieving separation are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing subsystem.

ADV_HLD.5-13 The evaluator ***shall examine*** the high-level design to determine that each subsystem is correctly categorised as either TSP-enforcing or non-TSP-enforcing.

27 In ADV_HLD.5-12, the evaluator confirms that the high-level design identifies each subsystem as either TSP-enforcing or non-TSP-enforcing. This is intended to permit evaluation effort to be concentrated on limited areas of the TOE that contribute to security.

28 In this work unit the evaluator confirms that categorisation has been performed correctly. This work unit may be performed in parallel with ADV_HLD.5-14.

29 The evaluator ensures that the subsystems that are required to function correctly in order to fulfill the TSP have been identified as TSP-enforcing (a mapping of SFRs to subsystems may be useful here).

30 Subsystems that do not need to function correctly (that is, in accordance with its specification) for the TSP to be upheld should be identified as non-TSP-enforcing. Evaluators should check that interference from non-TSP-enforcing subsystems either:

a) cannot occur, based on the information contained in the high-level design (e.g. because of the interfaces provided) or

b) is prevented by one or more protection mechanism.

ADV_HLD.5-14 The evaluator ***shall examine*** the high-level design to determine that it describes how separation is achieved.

31 Clear and effective separation is demonstrated by showing that the non-TSP-enforcing functions are incapable of violating the TSP. That is, the failure or anomalous behavior of a non-TSP-enforcing function can not result in a violation of the TSP.

32 Separation may be achieved through physical or logical means. For example, separation may be achieved through the grouping of functions into a subsystem, and then restricting the interfaces to that subsystem, thereby separating that subsystem from others. This model of separation may be implemented by various means in both hardware and software.

33 It is possible (for example where the TOE is simple and its sole purpose is security) that the TOE may contain only TSP-enforcing components.

34 The high-level design may also identify protection mechanisms that monitor and enforce separation, an example of such a mechanism would be kernel mode operations in an operating system.

35 A software-only TOE may have dependencies on the underlying operating system to provide separation mechanisms such as the above: these dependencies should be clearly identified.

ADV_HLD.5-15 The evaluator ***shall examine*** the high-level design to determine that the described separation is effective.

36 The evaluator should examine each subsystem in turn and review its relationship with and impact on the TSP-enforcing subsystems. This review must show that that the non-TSP-enforcing functions are incapable of violating the TSP.

37 A model of the high-level design could be exercised under a variety of scenarios to ensure that the chosen structure provides the required separation and independence of the TSP-enforcing subsystems. These scenarios will include the threats identified in the security target of the TOE.

38 The use of such a model should be an iterative exercise. For example the model can be enhanced to include weaknesses identified in later stages of the evaluation, to ensure that separation and independence are maintained.

ADV_HLD.5.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

ADV_HLD.5-16 The evaluator ***shall check*** high-level design to determine that it identifies the security functions that are implemented by TSF mechanisms.

39 If the ST contains no security mechanisms in the TSS (refer to ASE_TSS.1.4C), this work unit is not applicable and is therefore considered to be satisfied.

40 This may also be determined by tracing the SFRs which are implemented by a security mechanism back to the mapped high-level design subsystem.

ADV_HLD.5-17 The evaluator ***shall examine*** the high-level design to determine that it describes how the TSF mechanisms implement the security functions.

41 If the ST contains no security mechanisms in the TSS (refer to ASE_TSS.1.4C), this work unit is not applicable and is therefore considered to be satisfied.

42 The evaluator determines that the TSF mechanisms described in the ST are correctly described by the high-level design subsystem which maps to the mechanism. The evaluator determines this by examining the described interfaces and functionality of the mapped subsystem, and comparing them to the TSF Mechanism description in the ST to ensure that they agree.

1.3.2 ADV_HLD.5.2E

ADV_HLD.5.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.5-18 The evaluator ***shall examine*** the high-level design to determine that it is an accurate instantiation of the TOE security functional requirements.

43 The evaluator analyses the high-level design for each TOE security function to ensure that the function is accurately described. The evaluator also ensures that the function has no dependencies that are not included in the high-level design.

44 The evaluator also analyses the security requirements for the IT environment in both the ST and the high-level design to ensure that they agree. For example, if the ST includes TOE security functional requirements for the storage of an audit trail, and the high-level design stated that audit trail storage is provided by the IT environment, then the high-level design is not an accurate instantiation of the TOE security functional requirements.

45 The evaluator should validate the subsystem interface specifications by ensuring that the interface specifications are consistent with the description of the purpose of the subsystem.

ADV_HLD.5-19 The evaluator ***shall examine*** the high-level design to determine that it is a complete instantiation of the TOE security functional requirements.

46 To ensure that all ST security functional requirements are covered by the high-level design, the evaluator may construct a map between the TOE security functional requirements and the high-level design.