Australian Government
Department of Defence

Defence Signals Directorate

Australasian Information Security Evaluation Program

Low Level Design (ADV_LLD.2) - CC V2.2

Common Evaluation Methodology

15 February 2006

Version 1.2

Amendment Record

Version	Date	Description
1.0	29 August 2005	Release
1.1	18 November 2005	Releasable to AISEFs
1.2	15 February 2006	Added CC V2.2 to title

Table of Contents

1 SEMI-FORMAL LOW LEVEL DESIGN (ADV_LLD.2) 4

1.1 OBJECTIVES 4
1.2 INPUT 4
1.3 EVALUATOR ACTIONS 4
1.3.1 ADV_LLD.2.1E 4
1.3.2 ADV_LLD.2.2E 8

1 Semi-formal Low Level Design (ADV_LLD.2)

1.1 Objectives

1 The low-level design of a TOE provides a description of the internal workings of the TSF in terms of modules and their interrelationships and dependencies. The low-level design provides assurance that the TSF subsystems have been correctly and effectively refined.

2 For each module of the TSF, the low-level design describes its purpose, function, interfaces, dependencies, and the implementation of any TSP enforcing functions.

1.2 Input

3 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.

1.3 Evaluator Actions

1.3.1 ADV_LLD.2.1E

ADV_LLD.2.1C	The presentation of the low-level design shall be ***semiformal.***

ADV_LLD.2-1 The evaluator ***shall examine*** the low-level design to determine that it is presented using a semiformal style.

4 A semiformal specification requires the use of a notation which is explicitly defined. It may be based on a restricted subset of natural language. Alternatively, it may be based on accepted methodologies or diagrams, eg data flow diagrams, state transition diagrams or flow charts.

ADV_LLD.2-2 The evaluator ***shall examine*** the low-level design to determine that the semiformal notation used is capable of expressing features relevant to security.

5 For instance, both data models and data flow diagrams qualify as semiformal notations. However, a single notation, such as a data model or data flow diagrams, may not be capable of expressing every facet of the TOE security functions. In this case, a combination of notations could be used to provide a complete picture of the low-level design.

ADV_LLD.2-3 The evaluator ***shall examine*** the low-level design to determine that it contains all necessary informal explanatory text.

6 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.

ADV_LLD.2.2C The low-level design shall be internally consistent.

ADV_LLD.2-4 The evaluator shall examine the presentation of the low-level design to determine that it is internally consistent.

7 For guidance on consistency analysis see B.3.

ADV_LLD.2.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.2-5 The evaluator shall check the low-level design to determine that it describes the TSF in terms of modules.

8 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.2.4C The low-level design shall describe the purpose of each module.

ADV_LLD.2-6 The evaluator shall examine the low-level design to determine that it describes the purpose of each module.

9 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.2.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.2-7 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.

10 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 security functions.

11 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 supporting security functions, 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.2.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.2-8 The evaluator shall examine the low-level design to determine that it describes how each of the TSP-enforcing functions is provided.

12 The TSP-enforcing functions are those functions of the TSF that directly or indirectly enforce the TSP.

13 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 implementer. If the evaluator, using the implementer'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.2.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.2-9 The evaluator shall check that the low-level design identifies the interfaces to the TSF modules.

14 The low-level design should include, for each module, the name of each of its entry points.

ADV_LLD.2.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.2-10 The evaluator shall check that the low-level design identifies which of the interfaces to the modules of the TSF are externally visible.

15 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.2.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing ***complete*** details of effects, exceptions and error messages.

ADV_LLD.2-11 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 complete details of effects, exceptions and error messages.

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

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

18 In the case of external interfaces, the required description may be included in the functional specification and can be referenced in the low-level design without replication.

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

ADV_LLD.2.10C The low-level design shall describe the separation of the TOE into TSP enforcing and other modules.

ADV_LLD.2-12 The evaluator shall check that the low-level design describes the separation of the TOE into TSP-enforcing and other modules.

20 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 modules are contained in the TSF. Modules that cannot affect TSP enforcement are not part of the TSF.

1.3.2 ADV_LLD.2.2E

ADV_LLD.2.1E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_LLD.2-13 The evaluator shall examine the low-level design to determine that it is an accurate instantiation of the TOE security functional requirements.

21 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.

ADV_LLD.2-14 The evaluator shall examine the low-level design to determine that it is a complete instantiation of the TOE security functional requirements.

22 The evaluator ensures that all ST functional requirements are mapped onto applicable sections of the low-level design. This determination should be made in conjunction with the ADV_RCR.2 Semi-formal correspondence demonstration sub-activity.

23 The evaluator analyses the low-level design to determine that each TOE security function is completely described by the module specifications, and that there are no modules on which a TOE security function relies for which there is no specification in the low-level design.