Australian Government
Department of Defence

Defence Signals Directorate

Australasian Information Security Evaluation Program

Tools and Techniques (ALC_TAT.3) - CC V2.2

Common Evaluation Methodology

11 November 2005

Version 1.1

Amendment Record

Version	Date	Description
1.0	17 October 2005	Release
1.1	11 November 2005	Releasable to AISEFs

Table of Contents

1 COMPLIANCE WITH IMPLEMENTATION STANDARDS – ALL PARTS (ALC_TAT.3) 4

1.1 OBJECTIVES 4
1.2 INPUT 4
1.3 APPLICATION NOTES 4
1.4 EVALUATOR ACTIONS 4
1.4.1 ALC_TAT.3.1E 4
1.4.2 ALC_TAT.3.2E 6

1 Compliance with implementation standards – all parts (ALC_TAT.3)

1.1 Objectives

1 The objective of this sub-activity is to determine whether the developer and his subcontractors have used well-defined development tools (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results, and whether implementation standards have been applied.

1.2 Input

2 The evaluation evidence for this sub-activity is:

a) the development tool documentation;

b) the implementation standards; and

c) the provided implementation representation.

1.3 Application Notes

3 This work may be performed in parallel with the evaluation activities under ADV_IMP, specifically with regard to determining the use of features in the tools that will affect the object code (e.g. compilation options).

1.4 Evaluator Actions

1.4.1 ALC_TAT.3.1E

ALC_TAT.3.1C All development tools used for implementation shall be well-defined.

ALC_TAT.3-1 The evaluator ***shall examine*** the development tool documentation provided to determine that all development tools are well-defined.

4 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.

5 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

ALC_TAT.3.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.3-2 The evaluator ***shall examine*** the documentation of development tools to determine that it unambiguously defines the meaning of all statements used in the implementation.

6 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.x 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.

7 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.

8 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.x) 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.

9 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.

10 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.

11 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

ALC_TAT.3.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.3-3 The evaluator ***shall examine*** the development tool documentation to determine that it unambiguously defines the meaning of all implementation-dependent options.

12 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.

13 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).

14 At this level, the documentation of development tools used by third party contributors to the TOE has to be included in the evaluator's examination.

1.4.2 ALC_TAT.3.2E

ALC_TAT.3.2E The evaluator shall confirm that the implementation standards have been applied **for all parts of the TOE.**

ALC_TAT.3-4 The evaluator ***shall examine*** aspects of the implementation process to determine that documented implementation standards have been applied to all parts of the TOE.

15 This work unit requires the evaluator to analyse the provided implementation representation of the TOE to determine whether the documented implementation standards have been applied.

16 The evaluator should verify that constructs excluded by the documented standard are not used.

17 Additionally, the evaluator should verify the developer's procedures which ensure the application of the defined standards within the design and implementation process of the TOE. Therefore, documentary evidence should be supplemented by visiting the development environment. A visit to the development environment will allow the evaluator to:

a) observe the application of defined standards;

b) examine documentary evidence of application of procedures describing the use of defined standards;

c) interview development staff to check awareness of the application of defined standards and procedures.

18 A development site visit is a useful means of gaining confidence in the procedures being used. Any decision not to make such a visit should be determined in consultation with the overseer.

19 The evaluator compares the provided implementation representation with the description of the applied implementation standards and verifies their use.

20 At this level **it is required that the complete provided implementation representation of the TSF** is based on implementation standards, including third party contributions. This may require the evaluator to visit the sites of contributors. The evaluator may consult the configuration list required by the CM scope (ACM_SCP) to see who has developed which part of the TOE.

21 If the referenced implementation standards are not applied for at least parts of the provided implementation representation, this work unit fails.

22 Note that parts of the TOE which are not TSF relevant need not be examined.

23 This work unit may be performed in conjunction with the evaluation activities under ADV_IMP.