
UK IT SECURITY EVALUATION & CERTIFICATION SCHEME

UK CC Interpretation - UK/3.1/012

12 March 2007

Status: Endorsed for use in UK Scheme

Subject: Multi-platform TOEs

REFERENCES CC/CEM, Version 3.1, Revision 1, September 2006

Issue

1. This interpretation addresses the case where a sponsor of an evaluation wishes for the resultant certificate to confirm that the TOE operates in accordance with its Security Target on a number of underlying platforms that are not part of the TOE. The issue of specific concern is what level of information must be supplied to the evaluators, and what activities must they undertake, in this scenario.

Scope of Interpretation

2. In the context of this interpretation, a *platform* is any hardware, firmware or software underlying the TOE that is required by the TOE for its operation, but which is outside the scope of the actual TOE. In other words, this interpretation applies where the underlying platform is part of the IT environment for the TOE.
3. Whilst the focus of this interpretation is on handling multi-platform claims where the platform is part of the IT environment, it rests on certain premises (stated below) as to how a TOE is evaluated on an underlying platform in the first instance.
4. This interpretation describes a generic approach for addressing platform dependencies. Where additional technology-specific issues need to be considered (for example, where the platform is a Smartcard/IC), these will be the subject of other CC Interpretations or CC Supporting Documentation.
5. Note that this interpretation complements, and is intended to be consistent with, UK CC Interpretation UK/3.1/008 *Treatment of commercial hardware that is part of a TOE*.

Premises

6. This interpretation is based upon the following premises:
 - a. The extent to which a TOE evaluation must consider an underlying platform (within its IT environment) is limited to those aspects of the platform on which the security of the TOE depends. This relates both to the support the platform provides for the TOE's SFRs, and to the introduction of potential vulnerabilities into the TOE. Where it is evident (e.g. from testing) that problems exist in the platform which compromise the security of the TOE, these issues must be addressed.
 - b. The principle of reuse of TOE evaluation results may be applied, provided it is

supported by a sound rationale that justifies that the results are not impacted by changes to the underlying platform. This permits a cost-effective approach to evaluating multi-platform claims.

7. It should be clear from the first premise is that there is no requirement to evaluate the platform itself, nor indeed to evaluate any security functionality which it offers. In other words, insofar as the TOE evaluation is concerned, the platform is to be treated as a 'black box'. Assurance that the underlying platform does not undermine the security of the TOE will thus derive primarily from:
 - a. The search for public domain vulnerabilities (which must include a determination of whether there are any obvious vulnerabilities in the platform that could undermine the TOE's SFRs), and
 - b. Testing of the TOE on the platform (which will – at least indirectly - cover all aspects of the platform upon which the security of the TOE depends).
8. It is otherwise assumed that the platform operates correctly to provide its specified functionality and contains no vulnerabilities. Where further assurance in the platform is required, the TOE boundary must be extended to include the platform (or specific platform functionality) within its scope.

Interpretation

9. Where the TOE is certified for more than one platform, then this is a declaration that equivalent assurance has been demonstrated for the evaluation of the TOE on each platform.
10. It is first necessary for a TOE to be fully evaluated on one platform.
11. The default position is then that all evaluation activities must be repeated in full for each underlying platform. In practice, however, the effect of a platform change is limited in scope: to the need for resultant change in the TOE itself for operation on the platform, and to the behaviour of the TOE when run on the platform.

Impact Analysis and Rationale

12. A rationale provides an acceptable basis for justifying that the security assurance requirements for the TOE are met across the range of platforms claimed in the ST, without recourse to repeated evaluation activity in respect of every one of those platforms.
13. At the heart of the rationale is an impact analysis that identifies the impact of platform changes on the assurance in the TOE, and correlates each impact to those evaluator actions which would, if repeated, demonstrate that the required assurance exists. The impact analysis may be developed further to justify that the nature of an identified platform change is such that it is not necessary to repeat a given evaluation activity (which may therefore be waived for that platform). Any reliance on the platform will be detailed in the security architecture description, which will describe how the platform supports the TSF self-protection (e.g. hardware memory management, [CEM] paragraph 533) and prevents bypass of SFR-enforcing functionality (e.g. prevention of raw disk access supporting TSF access control, [CEM] paragraph 537). This description can be used as a basis on which to build the impact analysis, as it details what is expected of a platform to support the SFRs.

14. The sponsor is required to provide the following evidence in support of a multi-platform claim, in addition to the full set of CC deliverables (as directed by the security assurance requirements) in respect of the baseline platform referred to at paragraph 10 above:
 - a. All CC deliverables for the TOE where these are impacted by a platform variation.
 - b. Details of differences in the underlying platforms claimed.
15. Impact analysis and rationale is considered to be an evaluation activity that is based on the above evidence supplied by the sponsor. The sponsor may, nonetheless, choose to additionally provide the impact analysis and rationale; in this event the impact analysis and rationale will be independently evaluated by the CLEF.

Content of a Rationale

16. A rationale must relate all security assurance requirements (SARs) to platform variations. The nature of the platform variations must be detailed. However, where there are a number of platforms, there may be benefit in employing a generic platform variation entity (e.g. clock speed) to characterise variations of that type. This is not always possible for software platforms and platform variations should be considered in terms of variations in the platform interface specification (see paragraph 28 below).
17. Conceptually (if not actually in its presentation), this leads, in the first instance, to an impact analysis based on a 2 dimensional matrix, which comprises (say) rows for each of the SARs, and columns for the platform variation entities. For each cell in the matrix a justification must then be given to demonstrate that platform variation can have no effect on TOE assurance. There may, however, be scope for applying a given justification to a number of cells.
18. A third dimension to be considered is that of *assurance-related TOE entities*. These are TOE entities that relate to one or more SARs, and which may be affected in different ways by a given platform variation. Consideration of the impact of a platform variation on a SAR therefore requires that the impact on the different TOE entities be considered, where these relate to the SAR.
19. The table below identifies the assurance-related TOE entities that must, as a minimum, be considered by the rationale (where relevant to the SARs specified in the ST). The most commonly encountered potential impacts of a platform variation are also identified (to serve as illustrative examples). Further discussion and examples are provided at paragraphs 25-37 below.
20. The rationale must cover all roles the environment plays in the protection of the TSF, as identified in the TOE's security architecture description (provided for ADV_ARC.1).

Assurance-related entity	Possible impacts of a platform variation include:
SFRs	<ul style="list-style-type: none"> • The correct operation of a SFR may be affected. • There may be SFR-relevant changes to the TSF representations.
TSF protection mechanisms ¹	<ul style="list-style-type: none"> • The correct operation of a TSF protection mechanism may be affected • There may be changes to TSF domain separation, self-protection, non-bypassability a described by the security architecture.
Potential vulnerabilities	<ul style="list-style-type: none"> • Additional vulnerabilities may be introduced by a platform variation. • The rationale for non-exploitability of vulnerabilities, as declared for the TOE on the baseline platform, may be impacted.

21. The above list is not exhaustive. Firstly, it is limited in scope to the set of assurance requirements that are covered by [CEM]. Secondly, it explicitly addresses those assurance-related entities that are most commonly affected by platform variations; it is possible that other assurance-related TOE entities could be identified in specific cases.
22. It should be noted that, whilst the rationale must cover each assurance-related TOE entity, there is likely to be scope for applying a given justification to defined groups of TOE entities. For example, a rationale may address defined groups of SFRs (such as Security Audit) where the rationale applies to the group as a whole.
23. A verdict justification for an evaluation activity will be considered adequate if it is seen to follow logically, from a consideration of the impact of platform variation, that waiving repeated evaluation activity will not cause a reduction in assurance. Where appropriate the justification may reference other evaluation material. So, for example:
- A cell relating a set of FDP_IFF.2 (MAC) SFRs to clock speed may assert that this platform variation is of no concern, noting as justification that the vulnerability assessment activities uncovered no potential vulnerabilities that could take effect in this context.
 - A cell relating an FDP_RIP (Residual Information Protection) SFR to memory size would need to refer to the way in which mechanisms for clearing memory were implemented to justify that equivalent assurance exists for such a platform variation.
 - A cell relating to an FAU_STG.1 (Protected audit trail storage) SFR to the storage of audit records by the underlying operating system platform would need to refer to the way in which the audit data is passed to the operating system and to possibly to compatibility/regression testing of platforms to justify that equivalent assurance exists for platform variations in file handling.
 - A cell relating domain separation (a property of the TSF security architecture) to platform memory management functions could offer no justification (and thus require repeated evaluation activity) wherever the memory management interface specification varies.
 - A cell relating a covert timing channel to clock speed would need to refer to an analysis of exploitable bandwidths within a covert channel analysis.
24. The following sections consider further the impact on deliverables and evaluator activities for those assurance aspects most commonly affected by platform variations (i.e. ASE, ADV, ATE and AVA, as noted at paragraph 21 above). This should not be construed as

¹ As described in the security architecture.

ruling out the possibility of there being an impact on other assurance aspects (i.e., AGD and ALC).

Security Target & Certification Report

25. The Security Target must specify the platforms on which the TOE is to be evaluated, in order to provide a clear definition of the evaluated configuration (following CEM Annex A.6).
26. The Certification Report will specify the platforms on which the TOE was evaluated. It will identify the platform(s) on which the TOE has been fully tested. It will also outline both the nature of variation between platforms and the means by which assurance has been achieved on the other platforms.

ADV

27. The TOE itself may require modification for its operation on different platforms. These modifications may exist at any level of TSF representation according to the impact of the platform variation (i.e. they may exist in object code, source code, low-level design, high-level design or the functional specification) but will typically be made at the lower levels of representation. For example:
 - a. Porting a database from one operating system to another may involve the addition of new modules in the low-level design, in order to emulate functionality present in the first but not the second operating system.
 - b. Moving from a uniprocessor to a symmetric multiprocessor hardware architecture may involve modification of TOE source code.
 - c. The binary image of a TOE may contain a number of code paths, one of which is selected at run-time in order to correctly drive the hardware platform employed.
28. Platform variations assume significance when they involve a variation in the platform interface specification (e.g. involving a different chip instruction set, revised system call including addition error checking). More generally, even where the platform interface specification does not vary, a platform variation may be significant where the platform functionality varies (e.g. involving a change in clock speed, different web server). It is necessary not only to consider specific interface calls, but more generally to understand the role of the platform in contributing to or compromising the overall security of the TOE.
29. In the case where the TOE relies on a software platform, a developer test rationale must (as a minimum) consider the impact of platform changes on the *correctness* of the implementation of each of the SFRs and any cited TSF protection mechanisms. If the *correct* behaviour of the SFR/TSF protection mechanism can be argued, on a case by-case basis, as being independent of the behaviour of the platform interfaces (excluding improbable platform flaws), then the rationale can be accepted. Note that this has to be distinguished from the question of the *effectiveness* of the SFRs being undermined as a result of flaws in the platform, e.g. leading to corruption of data on which the TOE depends, or enabling bypass through inadequate access controls. This is reliant on the properties of the TSF design as described in the security architecture description, which details how the TSF provides self-protection.
30. The nature of the dependency of the TOE on the underlying software platform will vary from SFR to SFR, for example:
 - a. The TOE implements an audit generation SFR (FAU_GEN.1), where the dependency

on trusted timestamps (FPT_STM.1) is to be satisfied by the underlying platform. When the TOE compiles an audit record, it makes a call to a platform interface, and relies on the platform to return the date and time. If the platform fails to provide the timestamp in the expected format, then this could result in the SFR not being correctly implemented. Testing of FAU_GEN.1 will confirm that the platform upholds its responsibility, but it does not follow that this result can be extrapolated to other claimed platforms. However, if these other platforms are certified as meeting the FPT_STM.1 requirement, then this should be sufficient evidence for an impact analysis. (Note that the question of whether the timestamps can be trusted is one of effectiveness rather than correctness, i.e. to be addressed by the search for obvious platform vulnerabilities.)

- b. The TOE implements an audit generation SFR (FAU_GEN.1), but relies on the underlying platform to store the generated information (for retention and subsequent analysis). In this case, it seems reasonable to argue that testing the TOE on a single platform is sufficient to demonstrate that the TOE generates the appropriate audit data, and calls the platform interface in a correct manner. (Note that an evaluator can still hypothesise the existence of flaws in the underlying platform that would undermine the integrity of the generated audit data – but this is a matter to be addressed in the evaluator’s vulnerability analysis rather than by developer testing.)
 - c. The TOE implements access controls using security attributes that are stored by the underlying platform. This differs from the timestamp example in that the TOE does not expect the platform to return the attributes in a particular format – it is the TOE that handles the unpacking of the data retrieved by the platform. As far as the platform is concerned, it is no different to any other data it has to store and retrieve. The key requirement for the developer testing to verify is therefore that the TOE makes the platform interface call correctly, so that the stored data is correctly retrieved and interpreted appropriately. In this case, testing on one platform would appear to be sufficient to confirm that the TOE SFRs operate correctly across all platforms that implement the same specified interface. Evaluators are, once again, free to hypothesise vulnerabilities in the platform interface implementation that might cause the data to be corrupted, but this is not a matter for the developer’s testing to address.
 - d. The TOE is a DBMS, storing TSF data such as audit records and security attributes in the database files, as well as the data (information assets) it is required to protect. The DBMS relies on the underlying platform to store the database files without compromising the integrity of the data, and also to implement access controls in order to prevent the DBMS SFRs from being bypassed or otherwise subverted. The latter is clearly an issue for the evaluator’s vulnerability analysis. The former might be regarded as a correctness issue for the SFRs; however, it would be reasonable to argue that it would be excessively paranoid to require developer testing on other platforms simply to confirm that the underlying operating system can handle files in a proper manner. Evidence of compatibility/regression tests on the claimed platforms (not necessarily security tests, or test evidence that is in a form that complies with the ATE requirements) would strengthen the developer’s arguments here.
31. The evaluators must consider platform interface specifications at all assurance levels, in accordance with the requirements of the impact analysis and the deliverables supplied to support evaluation of the TOE. Increasing assurance levels will thus lead to greater rigour in the consideration of the TOE’s usage of platform facilities.

ATE

32. Repeated developer and evaluator testing in respect of platform variations will be required at all evaluation levels, unless a rationale can be given to demonstrate equivalent assurance. Whilst the fewer deliverables required at lower assurance levels may offer less evidence on which to base a rationale, testing requirements are less onerous at these levels. However, in the instance of the TOE relying on a software platform it is expected that the evaluators will **always** be required to independently validate the developer's test rationale by adopting a test strategy that involves running a sample of tests across all platforms claimed, with potentially specific tests targeted at specific platform variations.
33. Note that:
- Developer testing covers testing of SFRs and external interfaces. ATE_DPT introduces the additional requirement for testing to demonstrate the correct behaviour of any specific mechanisms cited in the TSF's security architecture description (ATE_DPT.*-1). These may need to be repeated on the different platforms to demonstrate consistent behaviour across the platforms..
 - Different (additional) tests may be required on different platforms to reflect possible variations in effect, as part of the evaluator's independent testing (ATE_IND).
 - Independent testing of platform variations should form part of the evaluators' overall strategy for testing a subset of the TSF. Note that, within the context of this requirement, it is acceptable for the evaluators to adopt a risk-based approach in which testing focuses on those variations that are judged to carry the greatest risk of a vulnerability arising from an implementation error. There are cases there a developer rationale may be strengthened if the platforms claimed are certified, or if there is supporting evidence of regression testing (albeit not addressing ATE requirements) for those platforms.
34. If changes are made to a TOE to accommodate different platforms, then some developer testing on those platforms is likely to be required. However, the developer's impact analysis of ADV changes may well be able to limit the testing, e.g. needing to cover only that subset of the TOE SFRs that are affected by the TOE changes.

AVA

35. Changes to TSF representations may, of course, introduce vulnerabilities. However, even where modification of the TOE is not required, a platform variation may still affect its security. For example:

TOE relying on a hardware platform:

- Moving a TOE, without modification, to a processor offering additional instructions, including privileged instructions accessible from user mode code, may introduce a vulnerability.
- Moving from one hardware platform to another may be unacceptable on account of a BIOS vulnerability.
- Whilst not requiring modification of the TOE, a change in hardware clock speed may be significant where covert timing channels are considered to constitute a threat.

TOE relying on a software platform:

- Moving a DBMS TOE, storing TSF data such as audit records and security attributes in the database files, as well as the data (information assets) it is required to protect to

another operating system platform may introduce a vulnerability of bypassing the underlying platform's access control to the database files compromising the integrity of the data.

36. The potential introduction of vulnerabilities into the TOE, as a result of platform variations, must be considered at all evaluation levels. Any potential vulnerabilities apparent to the sponsor must be addressed in updated vulnerability analysis deliverables. The evaluators will check for the introduction of vulnerabilities by reference to these and, as appropriate, any other deliverable.
37. Repeated penetration testing in respect of platform variations will be required at all evaluation levels unless a rationale can be given to demonstrate equivalent assurance, i.e. that such variations do not introduce new vulnerabilities, or undermine any arguments presented by the developer that known vulnerabilities are not exploitable. Penetration testing must address any potential vulnerabilities arising from the TOE's usage of platform facilities. As with independent testing (ATE_IND), it is acceptable for the evaluators to focus their penetration testing on those variations that are judged to carry the greatest risk of vulnerability.

Application under Assurance Maintenance

38. A sponsor may provide an impact analysis and rationale in support of an Impact Analysis Report submitted under an Assurance Maintenance process. This approach may therefore - subject to the approval of the CB, and established case law - be used to extend the certificate to new platforms under maintenance. It is likely that one prerequisite for this is that the impact analysis and rationale demonstrates that the impact of platform changes on assurance is *Minor*, such that assurance is maintained if the developers repeat their tests (or a subset thereof) on the new platforms (or a representative sample of those platforms).

A Note on Compatibility Testing

39. Where the developer operates a Platform Compatibility Testing (PCT) facility, with platforms which pass the PCT placed on a Platform Compatibility List (PCL), he may wish for this testing to contribute to assurance.
40. In principle this is acceptable. As a first prerequisite however, PCT must incorporate those tests required for the evaluation level by impact analysis and rationale, in accordance with paragraph 32 above (if the developer wishes to incorporate lower level tests then it must be justified by rationale that this gives equivalent assurance²). Essentially there are then two options in this area:
 - a. PCT test results (together with associated test plans, purposes and procedures) are supplied to the evaluators as a deliverable,
 - b. Subject to the agreement of the CB, the developer may cite the operation of PCT to justify platform changes as having a *Minor* impact on assurance, as part of an Assurance Maintenance process.

² For example, in some cases the rationale will be strengthened if the developer can supply evidence of regression/compatibility testing on the platforms claimed. In fact, it is reasonable to expect that the developer will (at least) have performed some such testing for all platforms claimed, and the provision of any such evidence (even if it is only made available during a site visit) should be encouraged.

41. The following points should be noted where it is desired that PCT form part of an Assurance Continuity process:
 - a. PCT must be controlled by well-defined procedures. The CB or CLEF must have independently checked these procedures and their application.
 - b. The PCT procedures must place constraints on the type of platform variations that can be accommodated by PCT alone.
 - c. Where PCT has been operated before the evaluation of the TOE the evaluators will need to confirm that the necessary assurance holds for all PCL entries back to the first PCL entry.
 - d. Where PCT is operated after evaluation of the TOE as part of an Assurance Continuity process, there will a need for a check of the application of its procedures to confirm that the necessary assurance holds for all latter PCL entries. This may be carried out by the CB (under Assurance Maintenance) or by a CLEF (as part of a subsequent re-evaluation).
 - e. Further details of this process must be agreed with the CB.
42. Where PCT is further extended to encompass the developer's partners, then assurance will also need to be demonstrated in respect of their contribution.

Rationale

43. There is obvious benefit to the sponsor from certification of a TOE on a number of platforms. This interpretation seeks to outline the issues involved in multi-platform evaluations, and to indicate a general approach acceptable to the CB.
44. Whilst the notions of platform variations and assurance related entities are reasonably straightforward to apply in the case of hardware platforms, some interpretation is needed where the platform is software; this interpretation thus expands on earlier versions of UK/012.
45. A clear distinction is drawn between what a developer needs to provide in terms of test evidence for ATE, and those issues to be addressed by evaluators under the AVA_VAN activity, where vulnerabilities in the underlying platform have to be considered.
46. Any multi-platform rationale will be independently validated by the evaluators as part of ATE_IND.2; the evaluators are not restricted in which platforms they choose to run their tests (or repeat developer tests) on.