Supporting Document Guidance

Collection of Developer Evidence

April 2012

Version 1.5

CCDB-2012-04-005

Foreword

This is a supporting document, intended to complement the Common Criteria version 2 and 3 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

Supporting documents may be “Guidance Documents”, that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or “Mandatory Technical Documents”, whose application is mandatory for evaluations whose scope is covered by that of the supporting document. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

Technical Editor: NLNCSA

Document History:
V1.5, April 2012 : Initial release.

Field of special use: None

Acknowledgments:

The governmental organisations listed below and organised within the Joint Interpretation Working Group contributed to the development of this version of this Common Criteria Supporting document.

France:	*Agence Nationale de la Sécurité des Systèmes d'Information*
Germany:	*Bundesamt für Sicherheit in der Informationstechnik*
Italy :	*Organismo di Certificazione della Sicurezza Informatica*
Netherlands:	*Netherlands National Communications Security Agency*
Spain:	*Ministerio de Administraciones Públicas and Centro Criptológico Nacional*
United Kingdom:	*Communications-Electronics Security Group(CESG)*

They also acknowledge the contribution of the work done by several smart card vendors, evaluation labs, and other companies organised within:

- *eEurope*
- *International Security Certification Initiative (ISCI)*

Table of contents

1 Background

1 The objective is to facilitate effective and flexible application of the Criteria. There is considerable flexibility in the form in which developers may supply deliverables as inputs to evaluation. This interpretation examines some of the alternatives that the developer may choose, and the ways in which the evaluator may respond while complying with the requirements of the criteria ISO/IEC 17025, and mutual recognition. It also identifies and considers cases where there may be a danger that evaluators undertake work that is strictly outside their scope.

2 The way the Criteria are phrased imply that the developer should supply specific documents containing each particular type of evidence. Normally it will be most efficient if that is the case; the effort required by the evaluator to review the evidence will thereby be minimised. However, there is no explicit requirement on the format of the evidence; only the information content is prescribed. In particular cases it may be more efficient for the developer to present the evidence in more diffuse form, which requires more substantial evaluator effort to marshal and review. Provided that this can be done objectively and impartially, this *collection* of evidence is completely proper and acceptable.

3 The emphasis is on the objective justification of evaluation verdicts from developer-supplied deliverables. Where objective justification is not possible, the work becomes *creation* rather than collection of evidence. The aspect of *creation* is presented in this document only to help the reader in making the difference with *collection*. This document does not describe how creation of evidence could be used in an evaluation.

4 This document makes the distinction between two different ways permitting to obtain the evidence required by CC:

- Documentation corresponding to classical approach: the developer delivers directly all the necessary information
- Information: based on existing developer documentation and completed by additional information written in collection of evidence reports (e.g. filled questionnaire)

2 Interpretation

5 This is an interpretation independent of the criteria.

6 The developer is responsible for providing the information required by the criteria. The evaluator may exceptionally collect some of this information provided that:

a) Evaluator contributions are fully endorsed by the developer

The information provided by the evaluator during collection process shall be accepted by the developer and integrated in the documentation configuration management of the TOE, i.e. registered as complementary evaluation evidence

b) Approval is given in advance by the CB

Before beginning the project, ITSEF and developer shall agree on the tasks which could use this method. The agreement shall be officially communicated to the certification body during the evaluation registration, which permits to inform and get an approval by the CB of the approach chosen by the evaluation. The CB is already informed that the tasks which can be evaluated with the help of this method are ALC, ADV and ATE. Nevertheless, for each evaluation, the evaluator shall inform the CB of what type of documentation will be provided directly by the developer, and what information will be collected by the evaluator.

c) The evaluator contributions are independently reviewed by other members of the evaluation team, and their review is documented in the ETR (or intermediary evaluation report)

Evaluation reports are already systematically reviewed, even in the classical approach, according to the standard ISO/IEC 17025. For the specific information produced by the developer during collection of evidence, attention of the reviewer is particularly focused on the verification that no creation of evidence has been done by the evaluator (only collection of evidence).

2.1 Collection of evidence

7 According to internationally agreed criteria and methodology, the developer must provide specified evidence but the format is not mandated. The evidence may be presented in a single document that addresses all the requirements of an assurance component, or the evidence may need to be collected from a number of documents. Collecting evidence from a number of separate sources and formats is legitimate evaluator work. It may be convenient for the evaluator to construct a working document that approximates the ideal developer deliverable, but it is not mandatory. The evaluator work must be limited to the objective *collection* of developer supplied material, rather than subjective *creation*, so that it remains repeatable, reproducible and impartial. A suitable test is whether any competent evaluator would obtain essentially the same result.

8 Objective collection of evidence is proper to the evaluators. It should not be considered as consultancy, and therefore does not need to be performed by an independent team.

2.1.1 Determining when collection of evidence can be useful

9 The collection of evidence method can be used by the evaluator to reduce iterations due to documentation changes. It is necessary to keep in mind that the method can permit to:

- Take into account developer practices (Fit the method with the practice) if the CC requirements can be covered
- Take into account evaluation limited workload, without impacting evaluation assurance level

10 A significant part of evaluation problems are due to documentation related iterations. That is to say, information is initially incomplete or inconsistent in documentation, even if the contents required by the assurance component can be finally verified after some documentation iterations. The goal of the method is to minimise these iterations on private/internal developer documents (it cannot be applicable to the security target or the guidance documentation of the product).

11 The second goal of this method is to base as much as possible the evaluation work on real documentation used by the developer and not documentation written for CC purpose only. The evaluator will use “Collection of Evidence” method to limit as much as possible some developer documentation written after the development.

12 Important note: this method targets documentation problems which do not cause a final evaluation verdict FAIL. Typically, a “documentation problem” issued by the evaluator permitting to conclude that a SFR is actually not implemented will not be solved by a “collection of evidence” method. This is the reason why the method guarantees the same evaluation level as the classical approach considering that the developer shall deliver directly all the information without the need for the evaluator to collect it.

13 Two different ways permit to obtain the evidence required by CC and shall be considered depending on evaluation cases: documentation and information (documentation completed by evidence collected, such as a filled questionnaire).

2.1.2 Determining the scope of the collection of evidence for a specific evaluation

14 The developer and the evaluator shall first make an assessment of the existing developer documentation in relation with the evaluation tasks ADV, ALC, ATE. Some initial documentation shall be available to the evaluator in relation with these evaluation activities; otherwise, it is clear that some aspects of the evaluation will not be covered. The level of information provided must give the evaluator and CB sufficient trust that the evaluation could succeed with a positive result.

15 This assessment shall take into account the targeted evaluation assurance level. Once this initial assessment is done, the evaluator concludes whether the targeted scope for collection of evidence usage is a priori acceptable, and informs the CB of what type of documentation will be provided directly by the developer (corresponding to which evaluation activities or parts of), and what information will be collected by the evaluator. The CB will be then able to approve or not the scope of collection of evidence usage.

16 The evaluation verdict finally guarantees the sufficiency of the initial set of documents delivered. Indeed, it results that the information which shall be directly provided by the developer (for strict conformance to the CC or for efficient understanding by the evaluator) has actually been provided; otherwise the evaluation verdict will be failed.

2.1.3 Preliminary activity: Training on product

17 This step is not strictly speaking part of the collection of evidence methodology. Nevertheless, it can be a benefit for the evaluator to quickly understand the context of the product environment and the context of the evaluation. Thus, the evaluator can take advantage of this training to determine whether he will be able to collect evidence during the evaluation. It also permits to understand TOE scope compared to the product.

18 During Security Target evaluation, the evaluator shall be trained on product functionality.

19 This training shall permit to:

- Improve the ST evaluation relevance
- Gain a functional knowledge of the TOE before starting FSP and guidance evaluation

20 During this training, the evaluator shall obtain:

- A description of the ST by the ST writer
- A TSFI description, but also a description of the other product interfaces which are not considered as “TSFI”
- A description of the tools supporting communication with the TOE. The developer shall deliver these tools to the evaluator
- Access to design information to be able to understand the product overall architecture (for evaluation assurance levels TDS component)

21 In case the evaluator would conclude at the end of this initial training that the developer’s input and the status of the documentation is not suitable or sufficient for ‘collection of evidence’, he would conclude that ‘collection of evidence’ is not feasible for the product under consideration. Consequently, the classical approach of the CEM would be preferred for the evaluation.

2.1.4 The collection of evidence in practice

22 Some assurance components cannot be concerned by the collection of evidence:

- ASE_xxx/APE_xxx: the ST/PP is a document being a basis for the whole evaluation, which will be often public. This document shall be complete and coherent as the evaluator will base the understanding of the evaluation on it.
- AGD_xxx: Guidance documentation constitutes part of the TOE delivered to the users. Deficiencies therefore constitute errors. It is not permissible for the Evaluator to make up for deficiencies.
- Work units of components that involve semi formal/formal method: semi formal and formal approaches are basically difficult to associate with incomplete documentation necessitating collection of evidence. Nevertheless, the collection of evidence can be used for parts of these components. For instance, a questionnaire can be used to understand the formal method used. But it will not be applicable to the formal model itself.

23 Due to the fact that the evaluator shall not create but rather collect evidence, the information provided by the evaluator will mainly have the form of a questionnaire constructed with open-ended questions that do not suggest the required answer. Indeed, the form of the questionnaire permits to focus the collection on the information actually required by the evaluator, corresponding to the information missing in the existing documentation (a preliminary work from the evaluator is necessary to determine which information is missing and to prepare the corresponding questions).

24 The evaluator shall make a clear difference between the information directly collected (i.e. the answers given by the developer) and his own analysis/comments directly linked to these answers. Indeed, as the same document can include both developer answers and evaluator analysis, it is fundamental to make a clear distinction between them. For example, if the questionnaire has the form of a table, the difference can be marked thanks to separate rows or columns developer answer/evaluator comment.

25 The CB will be informed when the interviews sessions occur and can decide to attend the sessions.

2.1.5 Evaluator contributions are finally endorsed by the developer

26 Once evidence has been completely collected, the evaluator delivers it to the CB and to the developer. The developer shall appropriate the information collected as it become complementary evaluation evidence. This evidence shall be included in the configuration management system of the TOE

27 The developer can decide to integrate the information collected directly in its own documentation to improve it for further evaluations and make easier reuse of it, but it is not required by the methodology.

2.2 Creation of evidence compared to Collection of evidence

28 The difference between objective collection and subjective creation of evidence is illustrated by considering the difference between an open-ended and a leading question to the developer. If the evaluator would make a definite hypothesis and would ask the developer for a yes or no confirmation, this falls on the side of creation of evidence, but an open-ended question that does not suggest the required answer falls on the side of collection of evidence.

29 A typical question corresponding to creation of evidence could be: “can you confirm that this SFR is implemented in this part of the design?”. Since the intention of the criteria is that developers should demonstrate familiarity with the IT features of the TOE, and that they have taken care with the security aspects, developer shall be able to answer to open questions corresponding to collection of evidence. The question corresponding to collection of evidence would be: “Can you indicate the part of the design where this SFR is implemented?”

30 The leading questions which are corresponding to creation of evidence are related to information directly linked to the evaluation criteria and to the verdict of the evaluation activity, such as leading questions related to design information, implementation of security measures in development environment, etc.

2.3 Small deficiencies

31 The evaluator may address small deficiencies in a developer-supplied deliverable by interviewing the developer and documenting his response, or by making hypotheses and requesting developer confirmation; however the evaluator should check the consistency of such input with other developer-supplied material. When doing so, the evaluator must supply a rationale, to be agreed by the Certifier, that the compensatory work is not excessive. Typically, the information and the rationale can be directly added in the evaluation report.

32 These small deficiencies shall be limited to some information such as careless mistakes, lack in a reference to a documentation which can be easily confirmed, etc. The difference with creation of evidence is the importance of information to be confirmed by the developer in relation with the criteria: the small deficiencies shall correspond to any incompleteness/inconsistency which does not have an impact on the verdict for the corresponding evaluation activity.

2.4 Examples of information which can be collected

33 The requirement for the developer to provide correspondence analysis does not necessarily demand the production of a tabular summary. If traceability is evident the evaluator may produce such a summary (if required) as part of the collection of evidence. On the other hand, if correspondences have to be inferred based on general similarities of the functions involved, then the work goes outside the scope of collection and correspond to creation of evidence.

34 The design supplied by the developer may be found to be incomplete in certain respects, for example it may not provide complete details of all modules. There is scope for evaluator collection of supplementary evidence from alternative sources, such as:

a) Other relevant design information.

This may include design documents for closely related TOEs, standard texts (e.g. on Unix or NT internals), as well as documentation relevant to the targeted version of the TOE which may provide a useful context (e.g. the functional specification).

b) Evidence in ETRs from previous evaluations of the TOE (i.e. involving an earlier version or a different variant).

Where the evaluators in a previous evaluation of the TOE have documented in detail their understanding of the internal workings of the TOE security, such evidence may assist the evaluators in gaining the required overall understanding of the internal workings of the TOE.

c) Developer presentations of particular aspects of the TOE security.

Developer presentations may help the evaluators to gain an overall understanding of particular parts of the TOE, for example how certain TOE security functions are implemented, or an overview of the internal workings of individual TOE subsystems. Such evidence may be used to complete the low-level design. Any information presented verbally which represents piece of evidence shall be documented by the evaluators and, any such input should be checked for consistency with other developer-supplied evidence.

d) Clarifications of specific technical queries from the evaluators, whether verbal or written (e.g. email).

Such evidence should be used to confirm the evaluator’s understanding of specific points of technical detail.

e) Evidence generated by the developer’s configuration management system.

Such evidence may be useful in helping to establish an accurate picture of the interrelationships between modules, e.g. call trees (identifying which modules depend on which other modules), use of global data structures by modules, and so on.

f) Module headers associated with the source code modules.

This will typically take the form of design evidence contained within comments in the source code modules or header files.

g) The source code itself, including any associated comments.

It is not anticipated that it would be practical to derive any substantial proportion of the detailed design from the source code itself, but it may be used to address particular questions of details, as comments within the source code may be.