



# Beyond Common Criteria's Mutual Recognition

Requirements for High Assurance Evaluations at EAL5, EAL6, and EAL7

# Table of contents

- 1 Introduction .....3
- 2 Scheme-specific procedures .....4
  - 2.1 United States (CCEVS / NIAP) .....4
  - 2.2 Germany (BSI) .....5
- 3 Requirements overview .....6
  - 3.1 Security Target.....6
  - 3.2 Design documentation .....6
  - 3.3 Implementation representation.....6
  - 3.4 Development life-cycle.....6
  - 3.5 Tests.....7
  - 3.6 Vulnerability assessment .....7
- A Comparison of assurance requirements from EAL4 through EAL7 .....9

# 1 Introduction

This overview is an attempt to guide IT product developers with a basic understanding of the Common Criteria approach [<http://www.commoncriteriaportal.org>] through the specific requirements found for high assurance levels, so that they can gain an understanding of the commitment they will have to make in order to successfully pass such an evaluation.

Evaluations according to the Common Criteria for Information Security Technology Evaluation are typically performed at one of the pre-defined, hierarchical Evaluation Assurance Levels, EAL1 through EAL7. Of these, EAL3 and EAL4 seem to be the most commercially relevant assurance levels - EAL4 is often considered the highest assurance level that can be achieved by moderately or very complex information technology products that have a sound architecture, but have not necessarily been designed with a certification in mind from the very beginning.

However, certain consumer types, such as the intelligence and defense communities, have a natural interest in higher assurance for the IT products they are buying; and certain hardware and software types, such as smart cards and real-time kernels for embedded devices, are based on a simplistic and straight-forward enough architecture that enables their evaluation at high assurance levels.

For this purpose, we will briefly discuss examples of Scheme-specific (procedural) requirements for higher assurance evaluations; provide a review of the basic concepts that are necessary to understand the assurance requirements defined in Part 3 of the Common Criteria; and then provide a table that details - for each of the assurance levels from EAL4 through EAL7 - a summary and comparison of the specific assurance requirements.

This document is current as of November 2009 and Common Criteria (CC) and Common Evaluation Methodology (CEM) Version 3.1 Revision 3.

## 2 Scheme-specific procedures

Only evaluations at EAL1 through EAL4 are mutually recognized by a number of countries throughout the world, based on the Common Criteria Recognition Arrangement. For example, the United States will recognize a certificate issued in Germany at EAL4, but not at EAL5. (An exception are the countries of the European Union which, based on a separate agreement, mutually recognize all evaluations regardless of their EAL.)

The fact that the higher assurance levels of the Common Criteria are not included in the international recognition arrangement between the participating nations throughout the world is – to a certain extent – based on the actuality that the higher scrutiny associated with these assurance levels is often sought in the context of national security. While it might be appropriate to rely on other nations to provide assurance for products that are going to be used in potentially sensitive, but not critical environments, it might be too risky to rely on a foreign nation for confirming the correct functioning of security in products that are being used in environments that protect more valuable assets.

With version 3 of the Common Criteria, achievements towards more comparability between evaluations at higher assurance levels have been made in that the CEM includes now jointly defined methodologies (work units) for most assurance components used in evaluations at EAL5 and higher, except for the components ADV\_SPM.1 and AVA\_VAN.5 that both are applicable to EAL6 and EAL7 evaluations.

Nevertheless, the schemes of the different certificate-issuing nations – while basically relying on common technical requirements – may or may not have different approaches and procedures for performing Common Criteria evaluations at higher assurance levels.

The following sections provide, on a high level, examples for some of the Scheme particularities in Germany and the United States that developers need to take into account when considering higher assurance evaluations. This is by no means exhaustive, and changes to scheme-specific procedures are sometimes made frequently – contact your evaluation lab for more details.

### 2.1 United States (CCEVS / NIAP)

#### NSA involvement

Policy Letter 15 specifies that the involvement of evaluation resources of the National Security Agency (NSA) may be required for evaluations that include assurance components above EAL4 (such as, ADV\_SPM.1 and AVA\_VAN.4). Basically, this means that – at the discretion of the NSA – they may either let your evaluation lab perform all the work, assign their own evaluators to work with your evaluation lab, or take over certain evaluation activities completely without the involvement of the lab. In addition, you will be required to provide the product, documentation, and evidence including source code, to NSA, along with training for the product. This also means that before entering an evaluation contract with your lab, a Security Target must be submitted to CCEVS so that NSA can determine their level of involvement.

#### Methodology

CCEVS cautions that an evaluation lab may have to develop (and provide for approval) a methodology for evaluation components not defined in the CEM and where such a methodology does not already exist within CCEVS.

#### Evaluation acceptance criteria

Independent of the EAL for the evaluation, the general acceptance criteria for the evaluation of products apply. This includes meeting one of NSA's Standard Protection Profiles or the U.S. Government Basic Robustness Protection Profile, and providing a Letter of Interest from a

Department of Defense, Intelligence Community, or other National Security customer. Check NIAP's website for details.

## **2.2 Germany (BSI)**

Methodology

AIS (scheme interpretation) 34 supplements the evaluation methodology for components not covered in the CEM.

## 3 Requirements overview

The table in the appendix basically summarizes the “input” requirements for evaluations at EAL4 through EAL7, arranged for easy comparison. Engineers should be able to use this overview in order to gain a basic understanding about what is required from an organization in order to take a product through an evaluation.

The following sections review some basic concepts that should be taken into account when comparing the requirements for the different assurance levels provided in the appendix.

### 3.1 Security Target

There is not much practical difference between Security Targets for different EALs – they serve as the reference for functional claims and the operational environment of the product that is subject to the evaluation. However, different schemes may or may not require that Security Targets are compliant with specific Protection Profiles defining a minimum set of functionality to be implemented, and a minimum level of assurance to be met.

### 3.2 Design documentation

The detail of design documentation expected at the different EALs, such as the Functional Specification and TOE Design Specification, vary. The assurance requirements in the CC differentiate between informal, semi-formal, and formal design styles. These styles are explained in further detail in the CEM, but a brief summary is provided here:

- Informal design: Specifications written in prose, in a natural language.
- Semi-formal design: Specifications based on a standardized presentation format (think UNIX man pages).
- Formal design: Specifications written in a notation based upon well-established mathematical concepts, providing formal proof for logical reasoning.

### 3.3 Implementation representation

For software products, the term “implementation representation” refers to the actual source code. While the only requirement on the product developer for the code inspection performed by the lab is essentially to provide the source code, there are significant differences in what this implies, depending on the EAL.

At EAL4 and EAL5, the lab will simply inspect samples of security function code. Starting at EAL6, the lab is required to inspect pretty much all of the source code. Meanwhile, the developer has to provide correspondence mappings for the code inspected by the lab into the design specifications, and needs to take into account that other design assurance requirements at EAL5 and higher, such as the requirement for a modular design in the ADV\_INT family, affects whether a specific implementation representation will be acceptable for evaluation or not.

### 3.4 Development life-cycle

Product developers are reminded that, typically and to an extent that may vary from scheme to scheme, all development sites that contribute to the product development (read: have write access to code, test cases, etc.) are subject to inspection (“site visits”) by the lab in order to verify that the required procedures are actually adhered to.

### 3.5 Tests

In addition to the testing that the product developer has to perform, the evaluation lab will conduct independent tests. These independent tests include verifying some of the developer's test results by repeating them, and devising additional tests for the security functionality of the product.

Up to and including EAL6, independent testing will repeat samples of developer tests and devise additional tests for samples of security functionality.

At EAL7, all developer tests will be repeated by the lab, and all security functionality must be completely addressed by lab-devised tests as well.

### 3.6 Vulnerability assessment

Product security evaluations following the Common Criteria aim to establish a certain amount of assurance that a product is free of security vulnerabilities. All evaluation efforts eventually lead towards a vulnerability assessment for the product, performed by the evaluation lab towards the end of an evaluation. When talking about Common Criteria assurance levels, it is therefore helpful to understand the vulnerability assessment objectives that are associated with the individual EALs.

The amount of effort required to perform vulnerability assessments increases with the EAL, and is influenced largely by two factors:

1. The amount of analysis performed, and the approach to performing an analysis, to identify potential vulnerabilities in the TSF increases from a "focused" analysis at EAL4 to a "methodical" analysis at EAL5-7.
2. The attack potential that threat agents are assumed to possess increases depending on the EAL. The TOE is expected to be resistant against attackers with an attack potential of "Enhanced-Basic" at EAL4, of "Moderate" at EAL5, and of "High" at EAL6 and 7.

In addition to these two factors, the amount of analysis necessary for an evaluation depends – obviously – to a significant extent on the amount and type of security objectives and security claims for the TOE, the techniques and mechanisms used to implement the functionality, and its complexity. For example, claims on information flow protection (such as, implementation of a mandatory access control policy) inherit a whole host of covert channel analysis and testing to be performed at higher evaluation levels.

#### On vulnerability analysis approaches

A focused analysis, in addition to taking into account obvious vulnerabilities that were encountered during evaluation activities or are in the public domain, will identify areas of potential vulnerabilities as well as areas of concern during the course of the evaluation. For example, when evaluating the module descriptions in the TOE design specification, the evaluator might develop a hypothesis for a potential vulnerability. Or, certain vulnerabilities might be known to be typical for the type of TOE being evaluated, and the design analysis and functional tests are not able to demonstrate that they are not existent in the TOE. Such potential vulnerabilities would then be picked up in the vulnerability assessment phase of the evaluation, where the evaluator attempts to identify an attack scenario that would allow to exploit the potential vulnerability. Before performing any further analysis, the evaluator validates that the attack scenario would be exploitable by attackers with an attack potential that the TOE should be resistant against. If this is the case, further analysis and/or penetration testing would be required to prove/disprove the existence of the vulnerability and its exploitability. Exploitability refers to the fact that, even if a vulnerability may exist in the product, specific deployment settings or other aspects might make it non-exploitable in the particular case of the evaluated configuration of the product, or the actual attack may require an attack potential larger than the one the TOE should be resistant against.

A methodical analysis takes this further. Based on an analysis of the security objectives postulated for the TOE in the ST, flaw hypotheses for all potential attack vectors are created. For example, if the ST contains an objective (and corresponding SFRs) for the TOE to provide encryption of certain data based on symmetric encryption schemes, it is naturally an area of concern that an attacker might be able to obtain the symmetric key that would allow access to this data. Flaw hypotheses based on the evaluator's knowledge of the implementation of the encryption function and on general types of attacks on key material will then be developed by the evaluator. All available evidence, such as architecture and design documentation, will then be consulted in order to identify attack paths in the TSF that would allow an attacker to reach this objective. Depending on the attack potential that would be required for the hypothesized ways for the attacker to obtain the key, further assessment and/or penetration testing will then be performed to prove or disprove that exploitable attacks exist.

#### On attack potential

The Common Evaluation Methodology, in Appendix B.4, provides informative, but widely accepted guidance on how to rate, or calculate, attack potential in the context of CC evaluations. This includes taking into account assumptions about the technical sophistication of attackers, their knowledge of the TOE design and operation, the time it would take them to identify and exploit a vulnerability, which kind of access to the system they would have available to perform an attack, and the equipment required. Following our example above, this would significantly influence whether a vulnerability assessment would consider only attacks to obtain the symmetric key that would be limited to exploit flaws, for example, in the user-accessible interfaces to the TSF identified by random testing, or whether side-channel attacks, for example a timing attack, using bespoke technical equipment, knowledge how to exploit such side channels, and inside knowledge of the product need to be taken into account.

#### EAL requirements

At EAL4, the lab will perform a **focused** vulnerability analysis of the TOE, based on all previous evaluation steps that lead to the identification of potential vulnerabilities in evidence provided as well as in the public domain. Penetration testing will be performed to verify the presence or absence of hypothesized flaws, and the resistance of the TOE to attackers with an Enhanced-Basic attack potential.

At higher assurance levels, the lab will perform a **methodical** vulnerability analysis of the TOE, based on all previous evaluation steps that lead to the identification of potential vulnerabilities in evidence provided as well as in the public domain. Penetration testing will be performed to verify the presence or absence of hypothesized flaws, and the resistance of the TOE to attackers with a Moderate (EAL5) or High (EAL6 and EAL7) attack potential.

## A Comparison of assurance requirements from EAL4 through EAL7

The following table provides a comparison of assurance requirements, synthesized from the assurance components that apply to EAL4 through EAL7. Its organization should be straight-forward to understand: For each of the assurance classes (e.g., ADV "Development"), the table lists for each EAL the applicable assurance components (e.g., ADV\_FSP.4 "Functional Specification") and then summarizes the requirements that can be found in Part 3 of the Common Criteria.

EAL4	EAL5	EAL6	EAL7
<b>Security Target (ST)</b>			
<p><b>Security Target (ASE_*.*)</b></p> <p>A full ST is required, describing the Target of Evaluation (TOE), the TOE Security Functions (TSF), the operational environment, Security Functional Requirements (SFRs), and other aspects of the evaluated configuration of the product.</p>	<p><b>Security Target (ASE_*.*)</b></p> <p>See EAL4.</p>	<p><b>Security Target (ASE_*.*)</b></p> <p>See EAL4.</p>	<p><b>Security Target (ASE_*.*)</b></p> <p>See EAL4.</p>
<b>Development</b>			
<p><b>Security architecture (ADV_ARC.1)</b></p> <p>A security architecture description at the same level of detail as the TOE design document required by ADV_TDS.3 is required. This description includes the security domains maintained by the TSF, and how the architecture prevent bypass of the TSF as well as tampering with the TSF during operation and startup.</p>	<p><b>Security architecture (ADV_ARC.1)</b></p> <p>A security architecture description at the same level of detail as the TOE design document required by ADV_TDS.4 is required. This description includes the security domains maintained by the TSF, and how the architecture prevent bypass of the TSF as well as tampering with the TSF during operation and startup.</p>	<p><b>Security architecture (ADV_ARC.1)</b></p> <p>A security architecture description at the same level of detail as the TOE design document required by ADV_TDS.5 is required. This description includes the security domains maintained by the TSF, and how the architecture prevents bypass of the TSF as well as tampering with the TSF during operation and startup.</p> <p>The preparation of such a document at the level of detail expected at EAL6 requires significant insight into the product internals on a (software) module level. Describing how modules interact to implement the TSF and protect themselves against interference and circumvention also requires a thorough understanding of the product's security architecture.</p>	<p><b>Security architecture (ADV_ARC.1)</b></p> <p>A security architecture description at the same level of detail as the TOE design document required by ADV_TDS.6 is required. This description includes the security domains maintained by the TSF, and how the architecture prevent bypass of the TSF as well as tampering with the TSF during operation and startup.</p> <p>See also EAL6.</p>

EAL4	EAL5	EAL6	EAL7
<p><b>Functional Specification (ADV_FSP.4)</b></p> <p>An informal FSP describes the TSF Interfaces (TSFI) , including their</p> <ul style="list-style-type: none"> <li>- purpose and method</li> <li>- all parameters</li> <li>- all actions associated with the interface</li> <li>- all error messages directly associated with the interface</li> </ul> <p>A tracing from the SFRs in the ST to the individual TSFIs must be provided.</p>	<p><b>Functional Specification (ADV_FSP.5)</b></p> <p>A semi-formal FSP describes the TSF Interfaces (TSFI), including their</p> <ul style="list-style-type: none"> <li>- purpose and method</li> <li>- all parameters</li> <li>- all actions associated with the interface</li> <li>- all error messages directly associated with the interface</li> <li>- all other error messages</li> <li>- and a rationale for all error message that are not associated with a TSFI</li> </ul> <p>A tracing from the SFRs in the ST to the individual TSFIs must be provided.</p>	<p><b>Functional Specification (ADV_FSP.5)</b></p> <p>See EAL5.</p>	<p><b>Functional Specification (ADV_FSP.6)</b></p> <p>A formal presentation of the FSP of the TSF, including their</p> <ul style="list-style-type: none"> <li>- purpose and method</li> <li>- all parameters</li> <li>- all actions associated with the interface</li> <li>- all error messages directly associated with the interface</li> <li>- all other error messages contained in the code</li> <li>- and a rationale for all error messages that are not associated with a TSFI.</li> </ul> <p>A tracing from the SFRs in the ST to the individual TSFIs must be provided.</p>
<p><b>Implementation representation (ADV_IMP.1)</b></p> <p>Provision of the complete implementation representation (for software TOEs, this is the source code) for the TSF, in the form used by the development personnel.</p> <p>A mapping between the TOE design description required in ADV_TDS and the sample selected by the lab must be produced.</p>	<p><b>Implementation representation (ADV_IMP.1)</b></p> <p>See EAL4.</p>	<p><b>Implementation representation (ADV_IMP.2)</b></p> <p>Provision of the complete implementation representation (for software TOEs, this is the source code) for the TSF, in the form used by the development personnel.</p> <p>A mapping between the TOE design description required in ADV_TDS and the entire implementation representation must be produced.</p> <p>For software TOEs, this means that the code for the TSF must be implemented on a modular basis, easy to read and with call paths that can be readily understood. The achievement of consistency with the TOE design description, as well as meeting the requirements for the TSF internals description, could - for example - be supported by diligently instrumenting code with comments that can be extracted by tools like JavaDoc or Doxygen, resulting in the (semi-) automated generation of module descriptions and their</p>	<p><b>Implementation representation evaluation (ADV_IMP.2)</b></p> <p>See EAL6.</p>

EAL4	EAL5	EAL6	EAL7
		<p>interface specifications in a standardized format. Preparation efforts increase on a linear basis with the amount of code and modules that the TSF implementation consists of.</p>	
<p><b>No requirement.</b></p>	<p><b>TSF internals description (ADV_INT.2)</b></p> <p>The design and implementation of the entire TSF must be well-structured. For software, this includes the expectation of a modular decomposition, adherence to coding standards, etc.</p>	<p><b>TSF internals description (ADV_INT.3)</b></p> <p>The design and implementation of the entire TSF must be well-structured and not overly complex. For software, this includes the expectation of a modular decomposition, adherence to coding standards, reduction of implementation complexity, etc.</p> <p>The TSF internals description is expected to provide rationale on how the developer achieved the goal of well-structured TSF internals with reduced complexity.</p>	<p><b>TSF internals description (ADV_INT.3)</b></p> <p>See EAL6.</p>
<p><b>No requirement.</b></p>	<p><b>No requirement.</b></p>	<p><b>Security Policy Model (ADV_SPM.1)</b></p> <p>An SPM must be provided that formally models all security policies expressed in SFRs that can be mathematically proven, demonstrating that the TOE cannot reach an insecure state.</p> <p>Development of a SPM requires significant knowledge in the areas of theoretical mathematics and formal proofs for security algorithms.</p>	<p><b>Security Policy Model (ADV_SPM.1)</b></p> <p>See EAL6.</p>
<p><b>TOE design description (ADV_TDS.3)</b></p> <p>The structure of the TOE must be described in terms of subsystems and modules. All subsystems must be described and mapped to modules. All modules must be described in terms of purpose and relationship to other modules. For SFR-enforcing modules, an interface specification must be provided. A mapping of the design to</p>	<p><b>TOE design description (ADV_TDS.4)</b></p> <p>The structure of the TOE must be described in terms of subsystems and modules. All subsystems must be described semi-formally and mapped to modules. All modules must be categorized as SFR-enforcing, -supporting, or -non-interfering. All modules must be described in terms of purpose and relationship to other</p>	<p><b>TOE design description (ADV_TDS.5)</b></p> <p>The structure of the TOE must be described in terms of subsystems and modules. All subsystems must be described semi-formally and mapped to modules. All modules must be categorized as SFR-enforcing, -supporting, or -non-interfering. All modules must be described semi-formally in terms of purpose and</p>	<p><b>TOE design description (ADV_TDS.6)</b></p> <p>The structure of the TOE must be described in terms of subsystems and modules. All subsystems must be formally specified and mapped to modules. All modules must be categorized as SFR-enforcing, -supporting, or -non-interfering. All modules must be described semi-formally in terms of purpose and</p>

EAL4	EAL5	EAL6	EAL7
the TSFI specified in the FSP must be provided.	modules. For SFR-enforcing and -supporting modules, an interface specification must be provided. A mapping of the design to the TSFI specified in the FSP must be provided.	relationship to other modules, and interface specifications must be provided. A mapping of the design to the TSFI specified in the FSP must be provided.  See also the notes on security architecture, implementation representation, and TSF internals description.	relationship to other modules, and interface specifications must be provided. A proof of correspondence between the subsystems and the FSP must be provided.
<b>Guidance documents</b>			
<b>Operational user guidance (AGD_OPE.1)</b>  For all user roles (administrators, users, etc.) the available functions and interfaces must be documented, including security parameters and secure values for interfaces, the description of security-relevant events, modes of operations, and security measures to be followed in the operational environment.	<b>Operational user guidance (AGD_OPE.1)</b>  See EAL4.	<b>Operational user guidance (AGD_OPE.1)</b>  See EAL4.	<b>Operational user guidance (AGD_OPE.1)</b>  See EAL4.
<b>Preparative procedures (AGD_PRE.1)</b>  Procedures need to be documented educating users how to securely accept the TOE from the developer, ensuring integrity of the product, and how to install the TOE and prepare the operational environment for operation in the evaluated configuration.	<b>Preparative procedures (AGD_PRE.1)</b>  See EAL4.	<b>Preparative procedures (AGD_PRE.1)</b>  See EAL4.	<b>Preparative procedures (AGD_PRE.1)</b>  See EAL4.
<b>Life-cycle Support</b>			
<b>Configuration Management (ALC_CMC.4)</b>  CM systems providing version and access control need to be used for all types of evidence ("configuration items") identified in ALC_CMS. Automated generation of the TOE must be supported (e.g., by having a build system in place that pulls source	<b>Configuration Management (ALC_CMC.4)</b>  See EAL4.	<b>Configuration Management (ALC_CMC.5)</b>  CM systems providing version and access control need to be used for all types of evidence ("configuration items") identified in ALC_CMS. The configuration items that comprise the TSF must be identified, and the CM system must be able to identify	<b>Configuration Management (ALC_CMC.5)</b>  See EAL6.

EAL4	EAL5	EAL6	EAL7
<p>code versions attributed to a specific label in the CM system into the build). Acceptance procedures for modified or new configuration items must exist.</p> <p>These measures need to be documented and followed.</p>		<p>dependencies between individual configuration items. Auditable version histories must be kept.</p> <p>Automated generation of the TOE must be supported (e.g., by having a build system in place that pulls source code versions attributed to a specific label in the CM system into the build). Acceptance procedures for modified or new configuration items must exist, and their appropriateness must be justified.</p> <p>These measures need to be documented and followed.</p>	
<p><b>Scope of CM (ALC_CMS.4)</b></p> <p>The following configuration items need to be under configuration management: the TOE itself, its implementation representation (source code), design and test documentation as well as test results, guidance documents, all other documents required as evidence for the evaluation, and security flaws and their resolution status in a bug tracking system.</p> <p>Configuration lists must be extracted and provided as evidence, identifying the exact version of each configuration item as it pertains to the evaluated version of the TOE.</p>	<p><b>Scope of CM (ALC_CMS.5)</b></p> <p>The following configuration items need to be under configuration management: the TOE itself, its implementation representation (source code), design and test documentation as well as test results, guidance documents, all other documents required as evidence for the evaluation, development tools and their documentation (such as compilers, etc.), and security flaws and their resolution status in a bug tracking system.</p> <p>Configuration lists must be extracted and provided as evidence, identifying the exact version of each configuration item as it pertains to the evaluated version of the TOE.</p>	<p><b>Scope of CM (ALC_CMS.5)</b></p> <p>See EAL5.</p>	<p><b>Scope of CM (ALC_CMS.5)</b></p> <p>See EAL5.</p>
<p><b>Delivery procedures (ALC_DEL.1)</b></p> <p>Delivery procedures must be in place and documented to maintain security when distributing versions of the TOE to the consumers.</p>	<p><b>Delivery procedures (ALC_DEL.1)</b></p> <p>See EAL4.</p>	<p><b>Delivery procedures (ALC_DEL.1)</b></p> <p>See EAL4.</p>	<p><b>Delivery procedures (ALC_DEL.1)</b></p> <p>See EAL4.</p>
<p><b>Development security (ALC_DVS.1)</b></p> <p>Physical, procedural, personnel, and other security measures must be</p>	<p><b>Development security (ALC_DVS.1)</b></p> <p>See EAL4.</p>	<p><b>Development security (ALC_DVS.2)</b></p> <p>Physical, procedural, personnel, and other security measures must be</p>	<p><b>Development security (ALC_DVS.2)</b></p> <p>See EAL6.</p>

EAL4	EAL5	EAL6	EAL7
described and employed at all development sites to protect the confidentiality and integrity of the TOE implementation and its development documentation.		described and employed at all development sites to protect the confidentiality and integrity of the TOE implementation and its development documentation. A justification of the appropriateness of these measures needs to be provided.	
<b>optional: Flaw Remediation (ALC_FLR.*)</b> Documented procedures of the support life-cycle for the product can be supplied, showing how reports of potential security flaws are accepted and investigated, tracked, and - if appropriate - fixed, and how customers are supplied with related information and patches.	<b>optional: Flaw Remediation</b> See EAL4.	<b>optional: Flaw Remediation</b> See EAL4.	<b>optional: Flaw Remediation</b> See EAL4.
<b>Life-cycle definition (ALC_LCD.1)</b> The development of the TOE must be based on a documented and structured life-cycle model, showing that appropriate roles (such as, developers and testers) are defined and appropriate checks and balances (such as, quality gates) are in place.	<b>Life-cycle definition (ALC_LCD.1)</b> See EAL4.	<b>Life-cycle definition (ALC_LCD.1)</b> See EAL4.	<b>Life-cycle definition (ALC_LCD.2)</b> The development of the TOE must be based on a documented, structured, and measurable life-cycle model, showing that appropriate roles (such as, developers and testers) are defined and appropriate checks and balances (such as, quality gates) are in place. Metrics must be in place to measure the quality of the TOE and its development, and results of measurements must be documented.
<b>Tools and techniques (ALC_TAT.1)</b> Development tools, such as compilers and programming languages, must be well-defined and documented, including their implementation-dependent options.	<b>Tools and techniques (ALC_TAT.2)</b> Development tools, such as compilers and programming languages, must be well-defined and documented, including their implementation-dependent options. They must be based on standards, i.e. well-accepted common practices and/or guidance set forth by demonstrated experts.	<b>Tools and techniques (ALC_TAT.3)</b> Development tools, such as compilers and programming languages, must be well-defined and documented, including their implementation-dependent options. They must be based on standards, i.e. well-accepted common practices and/or guidance set forth by demonstrated experts. Third-party providers supplying parts of the TOE must adhere to these standards as well.	<b>Tools and techniques (ALC_TAT.3)</b> See EAL6.

EAL4	EAL5	EAL6	EAL7
<b>Tests</b>			
<p><b>Test coverage analysis (ATE_COV.2)</b></p> <p>Correspondence between developer tests and the TSFI documented in the FSP needs to be documented, showing that all TSFI have been tested.</p>	<p><b>Test coverage analysis (ATE_COV.2)</b></p> <p>See EAL4.</p>	<p><b>Test coverage analysis (ATE_COV.3)</b></p> <p>Correspondence between developer tests and the TSFI documented in the FSP needs to be demonstrated, showing that all TSFI have been tested, and that tests exercise all parameters of each TSFI (including boundary and negative testing).</p>	<p><b>Test coverage analysis (ATE_COV.3)</b></p> <p>See EAL6.</p>
<p><b>Test depth analysis (ATE_DPT.1)</b></p> <p>Correspondence between developer tests and the subsystems defined in the TOE design specification must be demonstrated, showing that all of them have been tested.</p>	<p><b>Test depth analysis (ATE_DPT.2)</b></p> <p>Correspondence between developer tests and the subsystems and all TSFmodules defined in the TOE design specification must be demonstrated, showing that all of them have been tested.</p>	<p><b>Test depth analysis (ATE_DPT.2)</b></p> <p>See EAL5.</p>	<p><b>Test depth analysis (ATE_DPT.3)</b></p> <p>Correspondence between developer tests and the subsystems and all TSF modules defined in the TOE design specification must be demonstrated, showing that all of them have been tested. The analysis must also demonstrate that the TSF operates in accordance with its implementation representation.</p>
<p><b>Developer tests (ATE_FUN.1)</b></p> <p>The tests identified in the test coverage and depth analyses must be performed and test plans, specifications, expected and actual results must be provided.</p>	<p><b>Developer tests (ATE_FUN.1)</b></p> <p>See EAL4.</p>	<p><b>Developer tests (ATE_FUN.2)</b></p> <p>The tests identified in the test coverage and depth analyses must be performed and test plans, specifications, expected and actual results must be provided. An analysis of the ordering dependencies (dependencies between tests, install and de-install procedures, etc.) between test procedures must be provided.</p>	<p><b>Developer tests (ATE_FUN.2)</b></p> <p>See EAL6.</p>
<p><b>Independent lab tests (ATE_IND.2)</b></p> <p>The TOE, and a set of resources equivalent to those that were used for the developer tests, shall be provided for testing.</p>	<p><b>Independent lab tests (ATE_IND.2)</b></p> <p>See EAL4.</p>	<p><b>Independent lab tests (ATE_IND.2)</b></p> <p>See EAL4.</p>	<p><b>Independent lab tests (ATE_IND.3)</b></p> <p>The TOE, and a set of resources equivalent to those that were used for the developer tests, shall be provided for testing.</p>

EAL4	EAL5	EAL6	EAL7
<b>Vulnerability assessment</b>			
<b>Vulnerability assessment (AVA_VAN.3)</b> The developer provides the TOE, suitable for testing, to the lab.	<b>Vulnerability assessment (AVA_VAN.4)</b> The developer provides the TOE, suitable for testing, to the lab.	<b>Vulnerability assessment (AVA_VAN.5)</b> The developer provides the TOE, suitable for testing, to the lab.	<b>Vulnerability assessment (AVA_VAN.5)</b> See EAL6.