

***Common Criteria:
Optional Security Requirements
and Functions?***

***James Arnold/
Terrie Diaz***

28 September 2007

- Introduction
- Overview of Options
 - Options in Products
 - Options in Requirements
- Summary
- Recommendations
- Conclusions

Note that the presentation applies both to Common Criteria version 2.3 as well as version 3.1. The examples were drawn from version 3.1.

- The Common Criteria (CC) is about evaluating products, known as targets of evaluation (TOEs), in the context of specific requirements.
 - The TOE is confined to an “evaluated configuration.”
 - Many requirements apply to the entire TOE at all times.
- However, a TOE might support many configurations or modes of operation where, in effect, its security functions are dynamic.
- This presentation explores the potentially variable nature of TOEs and how that might be addressed in the context of CC evaluations.

- Options in Products
 - Most products have options to some degree, some of which have been historically accommodated in Common Criteria evaluations and others of which are commonly avoided.
- Options in Requirements
 - While many requirements in the Common Criteria do not accommodate options, there are some that directly support TOE options and others that less obviously support the possibility of options.
- Each of these topics is explored in more detail.

- Product options are generally considered to complicate the notion of “evaluated configuration.”
 - This is particularly true when the options serve to change the security functions, or behavior thereof.
- Product options often result in, sometimes difficult, choices about what is going to be included within the evaluated configuration for the sake of cost and time.
 - This is particularly true when the options result in combinatorial analysis or testing situations.

- Optional product components
 - Present or absent in their entirety
 - Might or might not include *security* functionality
 - Choices made
 - At the time of purchase
 - At the time of installation
 - At the time of use
- Examples
 - Licensed feature (e.g., IBM® DB2® LBAC)
 - Product-specific agents (e.g., for an intrusion detection system)

IBM and DB2 are registered trademarks of International Business Machines Corporation in the United States and/or other countries.

- Optional product functions
 - Enabled, disabled, or alternate behavior
 - Might or might not include *security* functionality
 - Choices made
 - At the time of installation
 - At the time of use
- Examples
 - Network services (NAT®, DHCP®, remote access, etc.)
 - Labeling services (e.g., Solaris® Trusted Extensions)
 - Audit function

NAT is a registered trademark of Neuberger Anlagen-Technik AG in the United States and/or other countries. DHCP is a registered trademark of MetalInfo, Inc. in the United States and/or other countries. Solaris is a registered trademark of Sun Microsystems, Inc. in the United States and/or other countries.

- Optional supporting components (in the environment)
 - Might or might not include supporting *security* functionality
 - Choices made
 - At time of purchase
 - At the time of installation
 - At the time of use
- Examples
 - Execution platforms (e.g., host for applications)
 - IT services (DBMS, LDAP, cryptographic library, etc.)
 - Clients (e.g., choice of Web browser)

- Effects of optional components and functions
 - Can add or remove entire security functions
 - Can augment or impair (i.e., change) existing security functions
 - Best case: no bearing on security functions
- Effects of optional environment components
 - Can serve to change TOE security functions
 - Can serve to change the TOE implementation
 - Best case: the TOE implementation and security functions remain unchanged
- In general, the range of options in the TOE and its environment can yield a wide range of possibilities.

- Add, remove, or change security functions
 - This type of variability represents a problem when determining and constructing appropriate requirements for the TOE.
 - Need to address any necessary security management control of potential changes within the requirements
 - Need to address the possibility of changes in security function behavior within the requirements
 - This type of variability can also yield a combinatorial problem where functions are interdependent, and their functions might potentially change relative to other changes.

- Change the TOE implementation
 - This type of variability will most likely yield a situation where each TOE implementation would need to be individually tested and, perhaps, analyzed, since the TOE behavior may have changed due to changes in its implementation to accommodate a different environment element.

- Regardless of any options that might be available for a given TOE, the Common Criteria requires the specification of specific requirements for each TOE to be evaluated.
- In many cases, the CC requirements do not accommodate TOE options.
 - Rather, they dictate that *the TOE Security Functions (TSF) shall* do something specific without any allowance for options or conditions that might be available in the TOE.
 - There are both direct and indirect exceptions, however.

- Requirements offering no TOE options
 - Requirements with no operations (i.e., assignment or selection) without a “*be able to*” caveat
 - Example: “FPT_PHP.1.1 The *TSF shall provide* unambiguous detection of physical tampering that might compromise the TSF.”
 - This example requirement leaves no room for any options that might serve to remove or impair the corresponding function.
 - If an optional component, for example, were to implement the indicated function, it would have to always be present and working in the evaluated configuration in order to fulfill this requirement.
 - In practice, while this feature may be useful to some users, it may be unnecessary for others.

- Requirements offering no TOE options (cont)
 - Requirements with limited operations (some assignments or most selections) without a “*be able to*” caveat
 - Example: “FAU_SAR.3.1 The *TSF shall provide* the ability to perform [selection: *searches, sorting, ordering*] of audit data based on [assignment: *criteria with logical relations*].”
 - Even though the requirement author can make a selection and complete an assignment, this example requirement leaves no room for options regarding the provision of the applicable review function.
 - If, for example, the TOE could support either searching or sorting, but not at the same time (i.e., they are mutually exclusive configuration options), this requirement could not be satisfied.

- Requirements offering limited TOE options
 - Requirements with less limited operations (many assignments) without a “*be able to*” caveat may offer limited or indirect TOE options
 - Example: “FPR_PSE.1.1 The *TSF shall ensure* that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].”
 - In this case, the requirement author could potentially caveat the users/subjects and/or the list of subjects/operations/objects in the applicable operations such that they are limited to when the option is enabled.
 - For example, users could be unable to determine the real user name bound to operations implemented within an optional component or by an optional feature so that the requirement remains satisfied, regardless.

- Requirements offering general TOE options
 - Requirements with a “*be able to*” caveat tend to offer flexibility
 - Example: “FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.”
 - In this case, the requirement is that the TSF has to be capable of providing reliable time stamps, but it doesn’t necessarily have to be able to do so all the time.
 - This might seem a poor example by itself, but FAU_GEN.1, which depends on FPT_STM.1, also has the caveat “be able to.” The implication being that the TSF must be able to generate reliable time stamps when it is also configured to generate audit events. As such, both audit and reliable time stamp generation might be configurable components or functions that need not necessarily always be present in order to fulfill their corresponding requirements.

- Requirements offering user-definable TOE options
 - Requirements with assignments where rules are to be defined offer flexibility to the requirement author to support TOE options
 - Example: “FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment for each operation, the security attribute-based relationship that must hold between subject and information security attributes].”
 - In this case, the rules can be defined such that variations in TOE components and TOE functions are accommodated. Even when the rule must apply to “all” subjects and/or objects, the notion of “all” isn’t necessarily static and would automatically adjust to any applicable changes in the TOE or its configuration.

- The previous examples are for requirement *elements*, while the Common Criteria presents requirements in terms of *components* which include one or more elements.
 - While a given element might allow flexibility, another element in the same requirement component might not.
 - Example: FAU_GEN.1.1 indicates “The TSF shall be able to...” while FAU_GEN.1.2 indicates “The TSF shall record...”. However, the second element applies to audit records generated in the first element, and if the first element doesn’t generate records, it doesn’t apply even though it is not present as a “*shall be able*” type of requirement.
 - For the most part, the requirement authors seem to have taken this into account.

- It is common for options to be offered for products when purchased, installed, and used.
 - As such, a given product could have many possible configurations, involving potentially different TOE implementations and security functions.
- The Common Criteria requirements have been designed either to offer no, limited, general, or user-defined flexibility in terms of supporting the potential for options available within a given TOE.
 - As such, the Common Criteria supports some flexibility with regard to TOE options when defining the “evaluated configuration.”

- While the Common Criteria (CC) requirements offer some measure of flexibility, there are certainly cases where the CC requirements do not facilitate options that TOEs can, and often do, offer.
- As a result
 - Product “evaluated configurations” are limited to a relatively small number of specific options
 - Products are being evaluated multiple times with different options (i.e., different “evaluated configurations”)
 - The CC requirements are being extended with alternate requirements that facilitate actual TOE options

- Products should be evaluated as close as possible to the manner in which they are expected to be used.
 - All TOE implementations should be addressed.
 - All TOE options should be addressed.
- Optional components and functions should be generally accommodated in evaluations.
 - There likely is a commercial reason those options exist.

Recommendations - Combinations



- Unfortunately, combinatorial problems have always been a difficult issue in evaluations.
 - When the TOE itself is different (e.g., Windows® vs. UNIX® build), it is likely that each different implementation would need to be tested separately. However, if the behavior is the same, the analysis would likely be the reusable, depending on the assurance level of the evaluation.
 - When the TOE implementation is the same and the supporting environment components differ, analysis and testing can be limited in cases where any potential differences can be rationalized as not being security-relevant.
 - When TOE component or function combinations are involved, if they are not security-relevant, there should be no particular issue. However, if security functions are affected, it is best when those functions are modular and it can be determined they don't affect other functions; otherwise, much more analysis (and perhaps testing) would be required to address the combinations.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark of The Open Group in the United States and/or other countries.

- When the options available for a given TOE allow changes in its security functions that cannot be accommodated by the requirements available in the Common Criteria:
 - The approach taken in the *Certificate Issuing and Management Components Family of Protection Profiles* (CIMCPP) could be used to, in effect, define multiple TOEs (or “evaluated configurations”), develop a superset of requirements, and then selectively assign each requirement to the applicable TOEs.
 - In effect, a single evaluation would then be performed that simultaneously addresses multiple TOEs with overlapping requirements.
 - The evaluated configuration could be defined inclusive of available TOE options, and any requirements that are not adequately flexible could be explicitly modified to accommodate the necessary flexibility.

- In the future, the Common Criteria should take into account that it is important to evaluate security functions regardless of whether they will be used by all TOE users.
- While it may be important that the TOE itself is protected while offering security capabilities to TOE users, it is not always the case that the TOE must protect itself, since that protection could be passed on to the TOE environment in some cases and in some regards.
- In general, the Common Criteria requirements should be more carefully crafted to accommodate flexibility that is necessary to fully support the way products can and are actually used.

Conclusions

- Historically, the “evaluated configurations” of many products have been too limited, largely to simplify the evaluation.
- As a result, products are often not being evaluated in the manner in which they are most often used.
- There are ways to broaden the scope of “evaluated configurations” using the existing Common Criteria requirements, though they are not necessarily simple.
- The Common Criteria should be revised to better accommodate optional capabilities in order to encourage the evaluation of products as closely as possible to their intent environments.
- Unfortunately, just as it must require substantive effort to implement a product in multiple environments, it will require more evaluation effort to evaluate the product in those environments.

James Arnold

SAIC Accredited Testing & Evaluation Labs
AVP/Technical Director

James.L.Arnold.Jr@saic.com

Terrie Diaz

SAIC Accredited Testing & Evaluation Labs
Quality Assurance Director

Terrie.L.Diaz@saic.com

<http://www.saic.com/infosec/cctl.html>