The Center for Information Systems
Security Studies and Research

Technical Report NPS-CS-06-007

Initial Documentation Requirements For a High Assurance System

Lessons Learned

Paul C. Clark, Cynthia E. Irvine, Timothy E. Levin, Thuy D. Nguyen,
David J. Shifflett, Donna Miller

February 2006

ACKNOWLEDGEMENTS

This material is based upon work supported in part by the Office of Naval Research and other government sponsors. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors

Author Affiliation:

Center for Information Systems Security Studies and Research

Computer Science Department

Naval Postgraduate School

Monterey, California 93943

Abstract

The Center for Information Systems Security Studies and Research (CISR) is working on a project known as the Trusted Computing Exemplar (TCX). This project is developing a high assurance computing component that will be evaluated at the Common Criteria (CC) Evaluation Assurance Level 7 (EAL7). The processes, documentation, source code, and other evidence to support the evaluation will be openly shared. Documentation is a substantial part of this evidence. Although the CC does state documentation requirements for each EAL, related requirements are often spread across multiple families, and no summarization of documentation requirements is provided. Therefore it was necessary to study the CC carefully to determine such requirements for EAL7. A long list of required documents was developed. However, the TCX project found that when starting from scratch there are particular documents, described herein, that are precursors to serious design work. In addition, it was learned that interpretations of the CC, and the occasional terminology translation were required.

1 Introduction

The Center for Information Systems Security Studies and Research (CISR) has been working on a project called the Trusted Computing Exemplar (TCX). The TCX project "will provide an openly distributed worked example of how high assurance trusted computing components can be built" [1]. One of the computing components that the TCX project will build is a small separation kernel that can enforce process and data separation. In addition, a reference trusted application will be built to use this kernel [2].

Assurance is a measurement of confidence that a system's security features function as specified, and the likelihood that the system does not contain maliciously inserted code that provides unspecified behavior. A high assurance system is one that can be trusted to store and process information of high value, while a low assurance system is one that should only be used to store and process low-value information.

The motivation for the TCX project is the fact that few high assurance systems have ever been successfully completed or evaluated, and of these, they have all been proprietary. Thus, it is extremely difficult for those new to information assurance to learn how to construct high assurance systems. An objective of the TCX project is to provide the information that will allow more organizations to consider building high assurance products. It is intended to remove the "mystery" of high assurance development through a worked example. The contributors to the TCX project have not undertaken this project as novices, but rather have many years of experience in the commercial sector in the area of high assurance product development.

The validation that a system is high assurance is provided via an independent third-party evaluation. A key aspect of a high assurance evaluation is the documented methodologies, standards, and processes that are used throughout the product lifecycle. This paper describes the lessons learned in the TCX project while developing documentation prior to the engineering phase of development of a trusted component.

It should be noted that the conclusions reported here reflect the fact that the development group is small, and that the product being built is relatively small. Larger projects may require modifications to these conclusions to best fit their needs and environment.

2 Common Criteria

The Common Criteria (CC) is an internationally recognized standard for security of computing products. It is divided into three parts: Part 1, Introduction and general model; Part 2, Security functional requirements; and Part 3, Security assurance requirements [3]. While Part 1 describes the overall approach to evaluation, Parts 2 and 3 provide the standards for the security features a product can have, and the assurance that those features behave as specified, respectively. The focus of this paper is the documentation requirements in Part 3.

The CC predefines seven different levels of assurance, known as Evaluation Assurance Levels (EALs), where EAL1 is the lowest assurance, and EAL7 is the highest assurance. The requirements of these 7 levels are carefully selected individual requirements from Part 3, where each higher EAL imposes additional constraints or additional new requirements beyond those of the adjacent lower level. Thus, it is recognized that EALs 1 through 4 are "low" assurance levels, while EALs 5 through 7 are "high" assurance levels. To provide the maximum benefit as an example, it was decided that the TCX separation kernel will be targeted for an EAL7 evaluation. The requirements for EAL7 drove most of the documentation requirements.

All the assurance requirements of the CC are divided into seven Assurance Classes, which can be thought of as different categories of assurance requirements "that share a common focus", e.g., configuration management and vulnerability assessment [3]. These classes are broken down further into Assurance Families, and finally into individual components, "which is the smallest selectable set of requirements" [3]. The individual CC requirements are found in the component descriptions. The seven different EALs have been defined by selecting components from the available families, as shown in Table 1, which is a reproduction from Part 3 of the CC. Each number in the table represents a component. Bold numbers represent additional requirements for that assurance family. Each component consists of from several paragraphs to several pages of requirements.

There is no simple reference, such as Table 1, to determine the overall documentation requirements for a given EAL. The documentation requirements are interspersed among all the other assurance requirements, with related requirements sometimes spread across multiple families. The only way to find the documentation requirements is to carefully read each component description that maps to the desired EAL, and merge the requirements together. Even then, some of the wording is vague and must be interpreted. In addition, the semantics of some of the terminology used in the CC was not the same as that of other standards with which the TCX team was familiar. In an attempt to minimize the risk of erroneous interpretation, the Common Methodology for Information Technology Security Evaluation (CEM) document was referenced from time to time [4]. The CEM document provides guidance to evaluators, and occasionally provides insight

into what the authors of the CC were thinking when they wrote the requirements. However, the CEM only provides guidance for low assurance evaluations (EAL4 and below). There is no corresponding guidance for high assurance evaluations, so some of the requirements for the high assurance EALs are not covered. An update to the CC during the development of the TCX requirements made this effort even more difficult.

Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration management	ACM_AUT				**1**	1	2	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	5	5
	ACM_SCP			**1**	**2**	**3**	3	3
Delivery and operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Development	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guidance documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGC_USR	**1**	1	1	1	1	1	1
Life cycle support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	1	2	2	2	2	2	3
Vulnerability assessment	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Table 1: Common Criteria Evaluation Assurance Level Summary [3]

3 Required Documents

After studying the CC components, it was determined that over 40 documents are required for an EAL7 evaluation. As overwhelming as that may sound, it was determined that only the following short list of documents need to be in place before development starts:

- Documentation Standards
- Life Cycle Plan
- Configuration Management Plan
- Configuration Management Procedures
- Configuration Items List
- Personnel Security Plan

- Physical Security Plan
- Software Development Standards

Failure to develop such documents prior to system development would render any attempt for a high assurance rating fruitless. They cannot be written post facto as an exercise to fulfill all high assurance requirements because the opportunity to have them contribute to system assurance is gone. Organizations that hope to receive a high assurance rating for a product must demonstrate that it was developed using industry best practices, and they must provide evidence that the practices were actually followed from the beginning of the product lifecycle. Thus, the documentation not only describes what must be done, but also describes how evidence shall be created and maintained for the evaluators.

During an evaluation, one of the first things that an evaluator will want to determine is whether the stated methodologies, policies and standards were good enough to qualify for the desired rating. Once that test is passed, the next thing an evaluator will want to see is proof that the organization actually adhered to them. Therefore, when writing any of the documents listed above, the author must continually ask the question: "How will I prove that I did all the things I said I was going to do?" In other words, what evidence will be needed to show that the stated methodologies, policies and standards were strictly followed from the start of development?

Another challenge for the developer is knowing when enough is enough. It is possible to go beyond the mark when considering how to meet all the requirements and do more than the minimum necessary for the desired rating. The evaluators will award the desired rating in such a case, but it will be at a greater cost to the developer than was necessary. Therefore, another question to keep asking is, "Is this too much?" It was valuable for the TCX documentation effort to have some team members keep asking this question.

It should also be noted that when the CC describes a documentation requirement, it does not mean that it must be met by a specific separate document – multiple documentation requirements can be met by a single document.

4 Documentation Standards

The first document that should be written is the Documentation Standards. This will define the look and feel of documents, the file format, etc. Otherwise, if left until later, the initial documents might need to be revised, wasting a significant amount of time and effort.

5 Life Cycle Plan

The next document that should be created is the Life Cycle Plan, because it provides the framework for how the product must be designed, built, upgraded, and potentially retired. Additionally, it describes the overall philosophy of development, and how all development activities fit together, especially with respect to the CC documentation requirements.

Among other things, all required documents must be described in the Life Cycle Plan, and when they are to be produced in the development process. Dependencies on other documentation should also be shown. It was found to be very helpful to break down the life cycle into phases, and then to break down the phases into activities. This allows the plan to show which items are required to start the activity, i.e., its inputs, and which items are produced by the activity, i.e., its outputs. The phases and activities defined for the TCX project are shown in Table 2.

Phases	Activities
Conceive	Product Definition
	Requirements Definition
Design	Product Design
	Detailed Design
Build	Implementation
	Testing, Composition and Validation
	Packaging and Delivery
Modify	Maintenance
Retire	Retirement

Table 2: Life Cycle Phases and Activities

As noted earlier, there may be a need to "interpret" certain CC requirements. Such interpretations should be documented in the Life Cycle Plan. One vague statement that caused much debate within the TCX project was the EAL7 requirement for a "standardized and measurable life-cycle model" [CC]. The words "measurable" and "model" do not seem to go together because a model is an abstraction and not a measurable item. In the end, this requirement was interpreted to mean that the "model must be simple enough to be able to identify (i.e., measure) the project's current status, with respect to the model" [5]. Having stated inputs and outputs for each activity makes it easy to determine the status of a project, with respect to the development model. Table 3 shows a list of the required CC documents for EAL7 and the activity that produces them.

In addition to the interpretation issues, there is the potential for inconsistent use of terminology between an organization and the CC. Such confusion must be dealt with in the Life Cycle Plan. One approach is to replace the CC terminology with the organization's terminology, but this can be very confusing for those who may refer to the CC, and may cause confusion among the evaluators. The terminology in the TCX documentation is consistent with CC usage, but to avoid confusion for internal personnel, strategic sidebar definitions were added to TCX documents.

The Life Cycle Plan is critical to the success of an evaluation. It must be written or revised with a good knowledge and understanding of all the CC requirements to ensure

that all the necessary documentation is generated at the right point in the development process.

6 Configuration management

After the Life Cycle Plan, the Configuration Management (CM) Plan should be developed. This may seem out of order, since at this point there would be at least two documents ready for configuration management, but the TCX documentation experience has shown that the Life Cycle Plan, which provides input to the CM Plan, should be written first.

The CM Plan describes the philosophy and high-level requirements for protecting configuration items after they have been developed. The plan needs to provide the framework so the following questions can be answered:

- How do you know that management approved development of the code?
- How do you know that the code submitted by a programmer implements the intended functionality?
- How do you know that the completed source code was not changed after it was submitted?

At some point in the development process, ideally in the Life Cycle Plan, the definition of a Configuration Item (CI) must be given. The CC requirements refer to the control of a system in terms of Configuration Items, but it "leaves the contents of the configuration item list to the discretion of the developer" [3]. In other words, the CC does not mandate how an organization decides to organize the collection of objects that constitute its product. For example, on one end of the spectrum a CI can be defined as a source file, and thus there would be many CIs. On the other end of the spectrum a CI can be defined as an entire product, so that there would be only one CI.

The disadvantage of having a lot of CIs is that it increases the administrative costs, because each time a change to any file is made, a review must be performed, and some amount of "paperwork" must be done to conform to the CM Plan. The advantage, however, is that there is a greater sense of control because changes are reviewed more often, and those changes may be small enough in scope to review them with confidence. The disadvantage of having a very small number of CIs is that there is a sense of having less control over the changes that are made to a system. An advantage is that fewer reviews are required. The TCX project selected an approach that is in between these two extremes, where a software CI is a subsystem, each document is a CI, third party tools are combined into one CI, etc.

Once the CM Plan has provided the framework and policies for controlled modifications of CIs, the CM Procedures can be written to provide the detailed steps for CM implementation. It should describe how new Configuration Items are submitted, and how changes to existing CIs are submitted to an approving body, such as a Change Control Board (CCB). In addition, it must define the materials that the CCB will review when considering requests. Finally, the CM Procedures need to specify the "paper trail" for all

changes in order to provide the necessary evidence that the items were properly managed in CM.

Activity	Documents Produced
Product Definition	Life Cycle Management Plan
	Product Definition
	Configuration Management Plan
	Configuration Management Procedures
	Configuration Items List
	Personnel Security Plan
	Physical Security Plan
	Development Standards
	Project Plan
Requirements Definition	Requirements Definition
	Acceptance Plan / Acceptance Tests
	Protection Profile (if necessary)
	Security Target (or in Detailed Design, depending on above)
Product Design	Formal TSP Model
	Functional Specification
	Formal Functional Specification
	Product Test Plan / Product Tests
Detailed Design	High-Level Design
	Formal High-Level Design
	Subsystem Test Plan / Subsystem Tests
	Low-Level Design
	Formal Low-Level Design
	Architectural Description
	Unit Test Plan / Unit Tests
	Covert Channel Analysis Plan
Implementation	Source Code
	Flaw Remediation Procedures
	Unit Test Results
	Administrator Guidance
	User Guidance
Testing, Composition and Validation	Covert Channel Analysis
	Code Correspondence
	Vulnerability Analysis
	Subsystem and Product Test Results
	Guidance Documentation Analysis
	Testing Analysis
Packaging and Delivery	Delivery Procedures
	Installation Procedures
	Integration Procedures
Maintenance	Assurance Maintenance Plan
	TOE Component Categorization Report
	Evidence of Assurance Maintenance
	Security Impact Analysis
Retirement	Retirement Announcement

Table 3: Document-to-Activity Matrix

After the CM Procedures have been written, the initial Configuration Items List can be produced. If each document were considered to be an individual CI, then there would initially be at least five entries, consisting of the documents described so far, including the CI List itself. At this point in the process, there is enough policy and procedure in place to submit these first five documents to the CCB for acceptance into and protection by the CM system.

7 Personnel Security Plan

A Personnel Security Plan must be written that describes the policies relevant to the people working on the project, to ensure the protection of the CIs. The policies need to address the following kinds of issues:

- Qualifications for assignment to the project (e.g., clearances, background checks, or citizenship).
- Training of users, with respect to the security requirements of the project, including both initial training and annual refresher training.
- How authorized users are managed on the project systems, such as the enabling and disabling of accounts.
- How audits are to be performed to ensure that the policies are being followed.

8 Physical Security Plan

A Physical Security Plan must be written that describes the policies relevant to the physical security of the CIs. The policies need to address the following kinds of issues:

- The physical security of development and CM servers.
- Integrity and confidentiality during transmission, such as on a network between development clients and servers.
- Separation of duties and roles such that CM privileges are not given to developers, and vice versa.
- Key and combination control.
- Backup policy and protection.
- Limitations on visitors.
- Audits.

9 Software Development Standards

Before software can be designed and implemented, engineering standards must be agreed upon. There are at least four parts to this document, though other standards can be added, as seen fit by the organization:

- The Review Process
- The Review Evidence
- Coding Standards
- Testing Standards

The review process describes how a CI shall be examined at critical stages during its development. For the TCX project, six review steps were defined for a CI during its development before it can be submitted to the CCB:

- Requirement Review
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Peer Review
- Acceptance Review
- Final Review

The Software Development Standards define what each review step must accomplish, and who is involved in the review. A critical part of this process is the review evidence, to show that the strict process was followed for the life of each CI.

Standardized coding practices must be documented. Each language has its own features that must be standardized, but the following provides a list that applies to all high-level languages:

- Commenting and Readability
- The use of Constants and Macros
- Header and footer requirements for files and functions
- Date formats
- Naming conventions

One of the review steps must verify that these standards were adhered to, such as the Peer Review.

Lastly, standards with respect to the testing of CIs must be stated.

If hardware components will be designed, then either a separate Hardware Development Standard needs to be written, or hardware and software standards can be combined into one Development Standards document.

10 Summary

Despite the fact that there are a large number of documentation requirements for a high assurance CC evaluation, our experience has revealed there are only eight documents that must be written before development can start. However, for a variety of reasons, when starting from scratch, these eight documents will take some significant time to develop, especially if one is not familiar with the CC. Careful thought must go into these documents because they form the whole framework for a high assurance environment, and provide the evidence that a product has earned a high assurance rating. The one difficult interpretation of the CC for EAL7 is the requirement for a “standardized and measurable life-cycle model” [3].

11 References

[1] Irvine, Cynthia E., et. al., *The Trusted Computing Exemplar Project*, Proceedings of the 2002 IEEE Workshop on Information Assurance and Security, pgs. 30-36, West Point, NY, June 2002.

[2] Nguyen, Thuy D., et. al., *TCX Project: High Assurance for Secure Embedded Systems*, 11th IEEE Real-Time Embedded Technology and Applications Symposium (presented as a Work-in-Progress), March 2005.

[3] Common Criteria for Information Technology Security Evaluation, version 2.2, January 2004, CCIMB-2004-01-001.

[4] Common Methodology for Information Technology Security Evaluation, version 1.0, August 1999, CEM-99/045.

[5] Life Cycle Management Plan, TCX000-LCMP00-000000, TCX Project, Center for Information Systems Security Studies and Research, Naval Postgraduate School, Monterey, California, September 28, 2004.