Maria Tulensalo

COMMON CRITERIA IT SECURITY STANDARD IN PRODUCT DEVELOPMENT PROCESS

Faculty of Electronics, Communications and Automation

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

Espoo, June 29th, 2010

Supervisor: Docent Kalevi Kilkki

Instructor: Master of Science in Engineering Marjut Rautavaara

A" Aalto-yliopisto

Teknillinen korkeakoulu

Common Criteria IT security standard in product development process

AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY	ABSTRACT OF THE MASTER’S THESIS

Author: Maria Tulensalo

Title: Common Criteria IT Security Standard in Product Development Process

Date: 29.06.2010	Language: English	Number of pages:8+85

Faculty of Electronics, Communications and Automation
Department of Communications and Networking
Research field: User-Oriented Telecommunications Technology Code: S-72

Supervisor: Docent Kalevi Kilkki

Instructor: Master of Science in Engineering Marjut Rautavaara

Information Technology Security is needed in both IT products and IT systems. One way to assure the secureness, is through the use of IT security standards.

In this thesis an international IT security standard called Common Criteria (CC) is examined in order to understand how it can be applied to a product development process, and what kind of benefits it brings to the process. This study begins by reviewing the basics of the IT security aspects, and by explaining the target of IT security standards. After that the content of the Common Criteria is examined in more details. The research was made based on a comprehensive literature research and a case using the Common Criteria evaluation assurance level 3.

The Common Criteria sets the basis for the whole life-cycle process of the product. Although implementing the CC requirements adds extra workload to the process, there are visible advantages for security related matters that could be left unnoticed without a compulsory requirement. The Common Criteria also receives wide international support and is considered as “the” de facto international standard for IT Security. However, its inflexibility mainly in terms of time and expenses has brought up a demand for developing it for a more dynamic IT standard.

Keywords: Common Criteria, security standard, security evaluation, information technology security, development process

AALTO YLIOPISTO

TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Maria Tulensalo

Työn nimi: Common Criteria -turvallisuusstandardi tuotekehitysprosessissa

Päivämäärä: 29.06.2010 Kieli: Englanti Sivumäärä: 8+85

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Tietoliikennetekniikan laitos

Tutkimusala: Käyttäjäkeskeinen tietoliikenneteknologia Koodi: S-72

Valvoja: Dosentti Kalevi Kilkki

Ohjaaja: Diplomi-insinööri Marjut Rautavaara

Tietoturvallisuutta tarvitaan informaatioteknologian (IT) tuotteissa ja järjestelmissä. Yksi tapa varmistaa tuotteiden turvallisuus on käyttää IT-turvallisuusstandardeja.

Tässä tutkielmassa tarkastellaan kansainvälistä IT-turvallisuusstandardia nimeltä Common Criteria (CC), jotta ymmärrettäisiin, kuinka sitä voidaan käyttää ja soveltaa tuotekehitysprosessissa, sekä mitä hyötyjä standardi tuo prosessille. Tutkielman alussa tutkitaan IT-turvallisuuden ja sen standardien perusnäkökulmia. Tämän jälkeen syvennytään Common Criteria –standardiin. Tutkielma pohjautuu kirjallisuuskatsaukseen sekä esimerkkiin, jossa käytetään Common Criterian arviointiolettamustasoa 3.

Common Criteria luo puitteet koko tuotteen elinkaarelle. Vaikkakin CC vaatimukset lisäävät työmäärää prosessissa, selviä hyötyjä turvallisuusasioihin on kuitenkin havaittavissa. Ilman ”pakollista vaatimusta” nämä turvallisuusasiat voisivat jäädä huomioimatta. Common Criterialla on myös laaja kansainvälinen tuki, ja sitä pidetäänkin tämän päivän merkittävimpänä yleisenä kansainvälisenä turvallisuusstandardina. Kuitenkin CC-standardin joustamattomuus ajan ja kustannusten suhteen on aikaansaanut uusia vaatimuksia sen kehittämiseksi dynaamisempaan suuntaan.

Avainsanat: Common Criteria, turvallisuusstandardi, IT turvallisuus, turvallisuusarviointi, kehitysprosessi

Preface

This Master's Thesis has been done at EADS Secure Networks as a part of my studies in Communicational Engineering department at Aalto University School of Science and Technology.

First, I would like to express my deepest gratitude to my technical advisor Simo Rinne for great guidance and feedback during the research process. I also want to thank my instructor Marjut Rautavaara for the interesting subject and for the opportunity to write this thesis. Special thanks to my supervisor docent Kalevi Kilkki for the constructive advices for the thesis.

I wish to express my sincere gratitude to all the helpful people at EADS for supporting me with valuable information. Lastly, I would like to thank my family for the support during my studies.

Espoo, June 29th 2010

Maria Tulensalo

Table of contents

List of Figures

List of Tables

Acronyms and terms

AES	Advanced Encryption Standard
CC	Common Criteria
CEM	Common Evaluation Methodology
CIA	Confidentiality, Integrity, Availability
CISSP	Certified Information Systems Security Professional
CTCPEC	Canadian Trusted Computer Product Evaluation Criteria
DES	Data Encryption Standard
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards
HW	Hardware
ICCC	International Common Criteria Conference
ISO	International Organization for Standardization
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria
MD5	Message Digest 5
OE	Operational Environment
OSP	Operational Security Policy
PP	Protection Profile
PSS	Public Safety and Security
SFR	Security Functional Requirements
SHA	Secure Hash Algorithm
ST	Security Target
SW	Software
SWOT	Strengths, Weaknesses, Opportunities, Threats
TOE	Target of Evaluation
TCSEC	Trusted Computer System Evaluation Criteria
TSF	TOE Security Functionality
TSFI	TSF Interface

1. Introduction

Our everyday life is more and more dependent on information technology (IT). That's why information technology security (IT security) has become an important factor as well, both for organizations and for individuals. Daily, all around the world, news about security flaws in computers and networks reaches ordinary citizens. "Customer's of Nordea as a victim of malware, the police investigating the subject" (Helsingin Sanomat, 2010). "Losses caused by the malware will be compensated" (Nordea, 2010). "Germany warns about Internet Explorer" (Digitoday, 2010). A malware program attacked a Finnish bank company Nordea and tried to lure customer's to give their personal bank account information with a fake log-in page. As a consequence, fifteen of Nordea customers lost a total of 50,000 Euros which Nordea then later compensated to the customers. In the other example, the government of Germany warned its citizens about the security flaws in Internet Explorer and advised people to use other web browsers instead. The warning was not considered as necessary by Microsoft, but a couple of days later after several reported attacks while using Explorer as a browser, Microsoft delivered a more secure version of their web browser.

Although information technology and its security as a means of computers, other electronic devices and networks has less than a hundred-year-old history, information and wanting to keep classified information secret itself has a history of hundreds of years. Julius Caesar (100 BC – 44 BC) used a "secret language" already over two thousand years ago to protect the security of the messages to his military (Lendering, 2010). He replaced each letter by three positions down the alphabet, as an example of substituting A for D, B for E and so on. From this "secret language" more advantaged and sophisticated methods of IT security have been evolved (Beissinger, 2006). Coming back to the 20th century and its history of IT security, one of the first publicized computer break-ins happened in August 1986. An intruder attacked the Lawrence Berkeley Laboratory in California. A computer engineer Stoll noticed a 75-cent discrepancy in accounting systems and started to investigate the matter. In the end, the intruder was found to be a

KGB-funded German who had been browsing sensitive databases, especially in military networks (Austin et al, 2009).

Today, a wide range of IT security methods and tools have been developed. We are all familiar with user names and passwords, firewalls and spam controls. However, it is said that one of the major problems in software security is the lack of knowledge about security among software developers (Rehman S. et al., 2009). Even if a developer has good knowledge about current software vulnerabilities, the general idea about the causes and measures which can avoid those vulnerabilities is limited. Researchers have shown the fact that most of the vulnerabilities arise in the design phase of the software development lifecycle (Rehman S. et al., 2009). One tool to this issue is to use international IT security standards and evaluations for the product being developed. Usually a security evaluation also covers the process of secure development which helps software developers to concentrate on the most important security aspects in the procedures. By evaluating and certificating a product, the developer can also state to the customers that there is certified proof of the product being secure, or that a product has been developed by using processes aiming to a secure solution.

IT security can be defined as protecting information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction (U.S Code collection, 2008). Other terms used synonymously with IT security are information security, data security and computer security. In this thesis, the term IT Security is being used by emphasizing the characteristics of security for information used and modified by the technology.

1.1. Scope, Research Methods and Questions

The premise to this thesis is to understand how the IT security standard called Common Criteria requirements can be fulfilled in a development project. At the same time the statement about lack of knowledge of IT security aspects in the development project can be examined. The goal is to find out whether by applying an international security standard, the IT security vulnerabilities in a product can be better avoided and whether the process is actually improved by the requirements given. In order to understand these aspects, the basics of IT security aspects and the purpose of IT security standards must be studied first.

The research methods applied in this thesis are a literature survey and a case using an IT security standard. The literature survey is used to understand the IT security aspects and security standards as well as how to apply those aspects in the development project. The case of an IT product development process will use the security standard Common Criteria (CC). Based on both the literature surveys and the example case, the results will be gathered to understand the pros and cons of applying security standard aspects in a development process.

The main research questions are defined as

1) How to apply IT security standard Common Criteria into a product development project?
2) What are the benefits and weaknesses of applying security requirements for a product in terms of different stakeholders?

1.2. Structure of the Thesis

The structure of the thesis is presented in Figure 1.

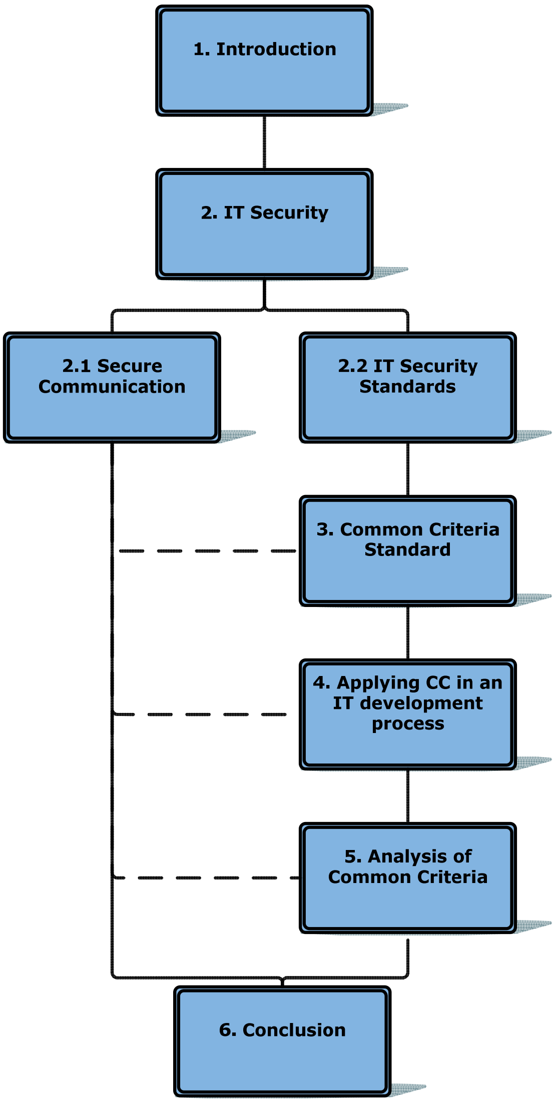

1. Introduction
2. IT Security
2.1 Secure Communication
2.2 IT Security Standards
3. Common Criteria Standard
4. Applying CC in an IT development process
5. Analysis of Common Criteria
6. Conclusion

Figure 1. Structure of the thesis

Chapter 1 introduces the history of information security as well as today’s information security and its vulnerabilities to the reader. Chapter 1 also introduces the reader the content and main research questions of this thesis. Chapter 2 explains more in detail the content of IT security and IT security standards. The secure communication section presents how secure communication can be divided into different aspects and how these aspects are managed. The IT Security Standard section investigates the purpose of IT security standards and briefly discusses a couple of different IT standards.

Chapter 3 concentrates on the chosen Common Criteria security standard. The chapter explains the generic terms used by the Common Criteria, presents the main parts of the requirements, and discusses different assurance levels that can be chosen.

Chapter 4 presents an application of Common Criteria to a development process. The application is presented in a case, where an IT product development project using waterfall process model is adapted to the Common Criteria evaluation assurance level 3 (EAL3).

Chapter 5 combines feedback from the case and former lessons learnt of other studies in analyzing the use of Common Criteria in a development project. The results are summarized in the end of chapter 5 in terms of its strengths, weaknesses, opportunities and threats (SWOT analysis). Finally, chapter 6 summarizes the main findings of the thesis and offer suggestions for continuation of the research.

2. Information Technology Security and Standards

This chapter is divided into two separate sections. The first section studies the information technology security. It begins with the introduction to IT security and its usage. The following chapters describe the different factors, security components, affecting IT security. The security components are examined in terms of threats and how these can be overcome by adding IT security features into the products and processes.

The second section discusses IT security standards. The section explains the background for information technology standards and what are the means by which customers want to make sure that a product is secure. The chapter also presents some of the well-known security standards available.

2.1 Secure Communication

The need and desire for information security varies between organizations and individuals (Conkling, W. R. et al., 2008). The motivation for a certain level of security depends on the situation. Dealing with highly sensitive information that could threaten an individual, a group of people or a nation, requires very strict security policies and security measures. Corporations have trade secrets and business processes that they want to keep out of publicity. Banks, medical organizations and states have records of personal information that could be used to steal personal identities.

In the annually held survey in the U.S by Computer Security Institute (2009), information security and information technology professionals in the United State answer questions about the security incidents they experienced and security measures they practiced. According to the last survey held (from the period of July 2008 to June 2009), the average losses due to the security incidents from cyber crime were $234 244 per respondent. The result of losses from annual Computer Security Institute's Computer Crime and Security Survey has come down from the highest results a couple of years ago, the amount being $350 424 in the year 2007. However, due to the nature of the report, there is speculation that everything is not told and the total amounts may be even higher (Richardson, 2007).

The original focus of IT security was on multiuser systems. In practice that meant that users had to be kept apart and unauthorized users had to be prevented from modifying systems software. Today, the focus is more on devices that function as the end systems in a network such as laptop computers and mobile handsets. The security issues rise from the fact that devices which are connected to a network may be attacked from untrusted nodes. (Gollmann, 2006)

2.1.1 Overview of IT security Aspects

IT security as a field of science is constantly changing and evolving. The security functions we have been using before might still be valid, but new methods must be developed all the time in order to understand new threats and counter possible new attacks. According to Ross Anderson (2001), the effect of the changes have moved IT security from a scientific discipline to an engineer discipline. Today’s security engineer is responsible for systems that evolve constantly and face a changing spectrum of threats. A significant part of the engineer’s job will be keeping up to date: understanding new attacks, learning how to use new tools, and keeping up on the legal and policy fronts. In addition, he has to understand the core disciplines of security functions as well as the basics of the management. However, the most important thing is to understand management of processes and communication with business people in order to achieve the consensus of technical aspects and changing business needs. (Anderson, 2001)

Also for the above named reasons, the study and examination of IT security aspects cannot be divided into clear segments that would be the same in most of the studies and research conducted. In this thesis, the viewpoint from John M. Kennedy T. (2009) is used by adding two more components, availability and non-repudiation, into the discussion. Figure 2 presents aspects of information security.

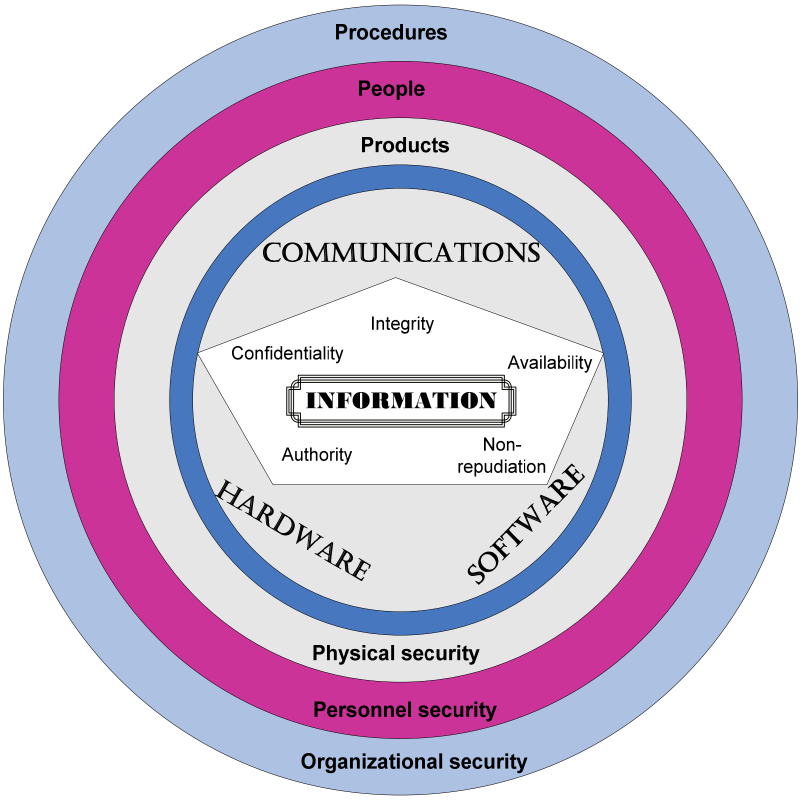


Figure 2. Information Security

Overall, there are three components which need security: products, people and procedures. The product aspect defines the physical security components, the people aspect defines the personal security components, and the procedures define the organizational security components. Information and its systems are composed of three main parts: software (SW), hardware (HW) and communications. In order to create security to the three first named components (the three outer circles of Figure 2), software, hardware and communications must be examined in terms of their properties. To the core of the properties belongs the confidentiality, integrity and availability (CIA) (Kennedy J, 2009). Also the fourth property, authority, is widely added to the basic group. In this thesis the fifth property, non-reputation, is being added to the research because of the case performed by this thesis and its importance in areas where recorded history must be available.

In the following sections each property is examined as an individual component. The property is discussed in terms of its meaning and main threats concerning the property. Also some solutions and tools are presented as countermeasures to the threats.

2.1.2 Confidentiality

Confidentiality means that only the sender and the intended receiver are able to understand the content of the transmitted message (Kurose, 2005). The threat for confidentiality is that an unwanted person could intercept the messages being sent. This is called eavesdropping, listening to and recording control and data messages on the channel (Kurose, 2005). Eavesdropping can be avoided by cryptography.

The basic principle of cryptography is that an original text called plaintext is converted into a coded (encrypted) form called a cipher text by a defined encryption algorithm. At the receiving end the cipher text is decoded (decrypted) and transformed back into plaintext. (Kurose, 2005)

There are numerous of encryption algorithm systems in use. However, encryption algorithms can be divided into two main categories: symmetric key systems and public key systems. In the symmetric key systems, the same key is used for encryption and decryption, and that's why it has to be kept in secret. In public key systems, also called as asymmetric encryption algorithms, two different keys are used for encryption and decryption. One of these encryption keys is public, so everyone can encrypt the message. The second key, however is called a private key, and has to be kept secret in order that only the receiver to decrypt the message. (Gollmann, 2006) Figure 3 presents the basic terms and encryption flow in a public key system (Asymmetric cryptography, 2007).

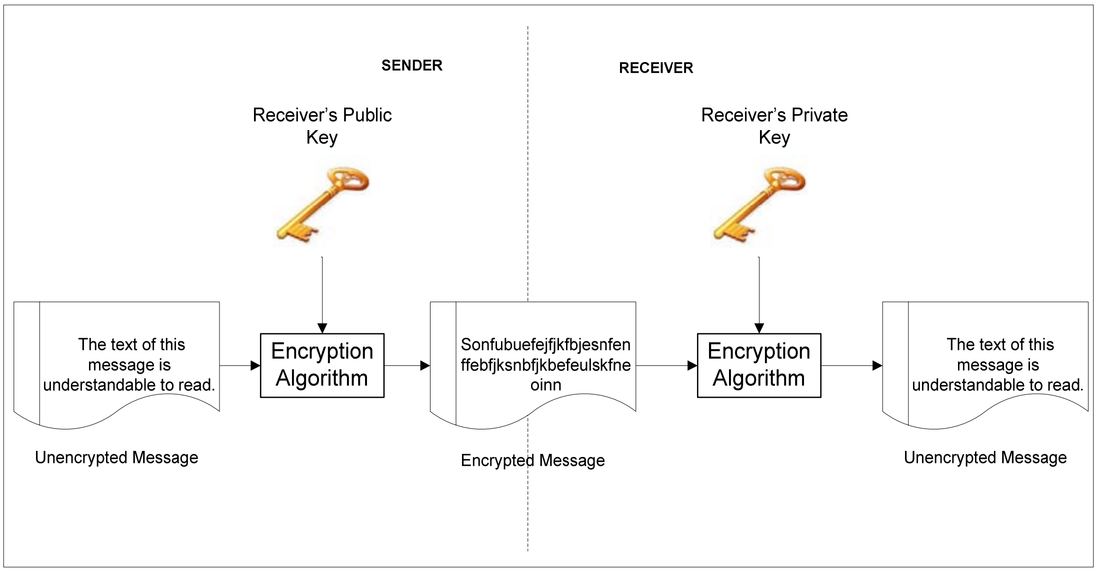

SENDER
RECEIVER
Receiver's Public Key
Receiver's Private Key
The text of this message is understandable to read.
Encryption Algorithm
Sonfubuefejfjkfbjesnfen ffebfjksnbfjkbefeulskfne oinn
Encryption Algorithm
The text of this message is understandable to read.
Unencrypted Message
Encrypted Message
Unencrypted Message

Figure 3. Asymmetric cryptography

The most widely used encryption algorithms are the symmetric Data Encryption Standard (DES), the Advanced Encryption Standard (AES) and the asymmetric RSA algorithm (Kurose, 2005).

2.1.3 Integrity

Integrity can be explained as making sure that everything is as it is supposed to be (Gollmann, 20006). Another definition of integrity is the assurance that the information is untainted (Interhack, CIA Security). Integrity does not speak for the accuracy of the information, only that the information sent is the same information as that which is received (Interhack, CIA Security).

The threat for integrity is manipulation. Manipulation can occur if data or control information is changed or deleted. Manipulation can be done by malware or also by an accident. The threat of manipulation can be avoided by integrity operations such as digital signatures and message digests. Integrity demands often also the same functions as defined under confidentiality and authority, such as cryptographic support and access control.

Digital signature is a technique used for signing documents and agreements digitally, which must fulfill three requirements. It must be verifiable, nonforgeable and nonrepudiable. In other words, a digital signature must prove that the signature is really made by the right individual (verifiable), and only by him (nonforgeable), and the signature has not been altered afterwards (nonrepudiable) (Kurose, 2005). According to Anderson (2001), the signature on the message can be created only by one individual, but can be checked by anyone.

Digital signature schemes can be either deterministic or randomized. In a deterministic scheme computing a signature always gives the same result, but in a randomized scheme the result is always a different, such as imitating handwritten signatures. A digital signature is, therefore a special type of asymmetric cryptography. The method of asymmetric cryptography is explained in the previous section ‘Confidentiality’. (Anderson, 2001)

A hash function, also called a message digest, is a function that makes it possible to check the integrity of the message. Once again, it is important to verify that the sender of the data is as claimed, and that the transmitted data has not been changed after creating and signing the data. A hash function is a simpler and computationally cheaper way to check the integrity than by encrypting the message, as full message encryption is a heavy operation to perform. In cases where the whole message does not need to be encrypted, message digest is efficient because only the authentication of the signature and the check of unaltered message are needed. (Kurose, 2005)

A hash function works technically the same way as checksums. Basically, the hash function algorithms take a message and compute a fixed-length "fingerprint" of the data called a hash function. This "fingerprint" is inserted into the message and when the recipient gets the message, he has to calculate the message's hash and compare it with the hash in the message. If the "fingerprints" from the message and from the hash fail to match, then it is clear that either someone has either deliberately, or accidentally, altered the signature or the message. (Kioskea, 2008)

The hash function verifies the fingerprint to match the received message but not that the message was sent by the person claiming to be the sender. To guarantee this, the sender has to encrypt the hash using his private key and send it to the recipient. In other words referring to the letter mail, the sender signs the seal and sends the seal to the recipient. When the receiver receives the message, he decrypts the seal with the sender's public key and compares the hash obtained with the hash function to the hash received as an attachment. Figure 4 shows a simplification of the hash function used with encryption from the sender to receiver. (Kioskea, 2008)

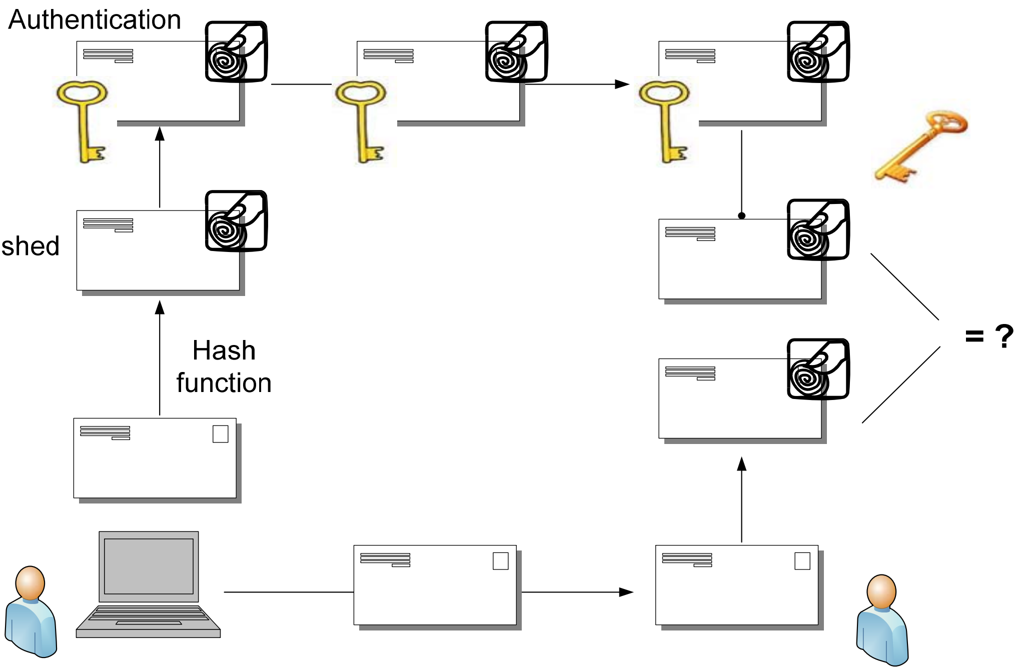


Figure 4. Hash function encrypted

The most common hash function algorithms used are Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) (Anderson, 2001).

2.1.4 Authority

Authentication is used so that both the sender and receiver can confirm the identity of the other party and so that only the authorized users of the system can access it. The threat behind authentication is that someone else may be impersonating the other party or someone else is listening in on the communication. A person between the actual communication parties listening illegally is called the man-in-the-middle. (Kurose, 2005)

One part of authority is access control. To protect sensitive information, the access to a database, such as those containing bank accounts, must be controlled. Some databases may also be available for a number of people but the need of actions made by these individuals may want to be varied (e.g. read and write rights). By giving different people or groups' different access and operation rights, access control can be achieved. (Gollmann, 2006)

The most common way to provide access control is to use a user ID and password. In our every day IT systems (access to computers, e-mails, Internet application etc.) the user ID is typed to identify the user and the password to ensure that the user is who he claims to be (authentication). In the simplest form, the password is compared to the password entered beforehand. (Koskinen, 2001)

The more sophisticated user ID - password methods use challenge-response authentication. In this method one party presents indirect questions (challenges) and the other party answers to those (responses). The challenge-response authentication is usually done by a one-way hash using random numbers or one-time passwords using a list, like logging in to a netbank account. (Koskinen, 2001)

The most critical network elements needing more protection for sensitive information use strong authentication. Strong authentication, also referred to as two-factor authentication, uses two out of the three proofs mentioned: something known (e.g. a password), something possessed (e.g. a credit card) and something unique about the appearance of a person (e.g. a fingerprint). (RSA Security, 2010)

The authentication process is usually done before starting the actual communication. A widely used practice in IT security literature is to use person A, Alice, and person B, Bob, as the communication parties when giving examples of IT security practices. If an intruder is in the picture, the person is usually called Trudy. Figure 5 presents an example of an authentication process working correctly (with no intruder) using an asymmetric form of authentication. (Kurose, 2005)

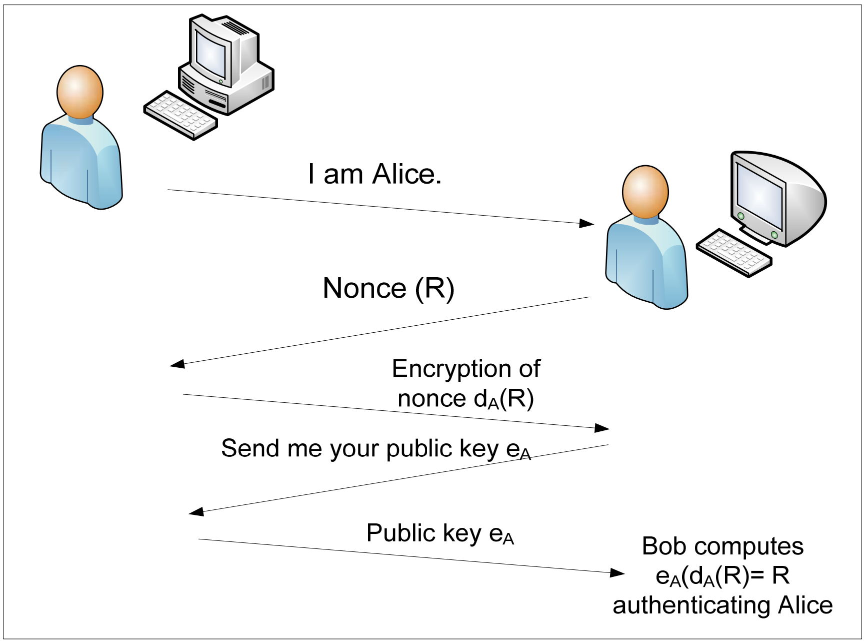

I am Alice.
Nonce (R)
Encryption of nonce $d_A(R)$
Send me your public key e_A
Public key e_A
Bob computes $e_A(d_A(R)= R$ authenticating Alice

Figure 5. Asymmetric authentication

In the beginning of the conversation, the authentication of the communicating party Alice is taking place. Alice sends a message to Bob telling who she is which is contained in the message “I am Alice”. Once Bob receives the message, he chooses a nonce and sends it to Alice. A nonce is a number that a protocol will use only once in a lifetime in order to avoid a threat that an intruder could intercept the password and use it later on when pretending to be Alice. Alice uses her private key, d_A, to encrypt the nonce and sends the resulting value $d_A(R)$ to Bob. Alice is the only one that knows her private key, meaning that no other person can generate the resulting value. When Bob receives the $d_A(R)$, he applies Alice's public key, e_A, to the received message. Thus, Bob computes the nonce, R, and if that matches with the nonce Bob sent to Alice, the authentication of Alice is successful.

2.1.5 Availability

Availability can be defined according to the ISO 7498-2 (International Organization for Standardization) standard as the property of being accessible upon demand by an authorized entity (Gollmann, 2006). In plainly speaking, the system must be available always as wanted. The threat against availability is denial of service. Denial of service can be achieved by over flooding or by using a malicious program.

With the aid of firewalls, authorization procedures and by defining access rights, high enough availability and access control can be achieved. According to Kurose et al. (2005), a firewall is a combination of hardware and software that isolates an organization's internal network from the wider Internet, allowing some packets to pass and blocking others. A firewall allows a network administrator to control access between the outside world and resources within the organization's own network by managing the traffic flow to and from resources. Authorization procedures were explained more in detail in the previous section.

In critical IT systems such as Public Safety and Security (PSS) networks, availability is a very important factor. According to Vargas (2000), the authorities want to be sure that in any case certain users will get service, because congested situations can jeopardize human lives or have a high economic impact. The capacity of the network is limited (e.g. because of the financial reasons), so availability is assured by building other solutions. Different priorities, pre-emption and subscriber classes make sure that the most important users can, at any time, have access to the system. In the case of an availability problem, a lower priority user action can be released to enable enough availability to a higher priority user. In the case of fallbacks in a network element or in a transmission line, different disaster recovery functions assure that the important parts of the network still work. (Rinne, 2010)

2.1.6 Non-repudiation

According to Gollmann (2006), non-repudiation services provide unforgeable evidence that a specific action occurred. This service enables the person in a communications session to prevent another party in the session from denying having taken a particular action. By non-repudiation services the discussions about having sent or received a message or operations performed by a user can be proven (Dent et al., 2004). One of the important users of non-repudiation is rescue teams such as the police and fire brigade. The orders given in critical situations must be afterwards possible to check and proven in case something has not gone as planned. In other words, the threat of non-repudiation is inability to provide evidence of occurrences.

Non-repudiation can be divided into two main groups. Non-repudiation of origin defines that the recipient of a message is provided with the means that the originator of a message cannot deny sending the message. Non-repudiation of delivery defines that the sender of the message is provided, with the means to prevent the recipient of the message from denying having received it and having recognized its content. (Dent et al., 2004)

There are various methods to assure the non-repudiation. Both the symmetric and asymmetric cryptography can be used for non-repudiation. These to forms of cryptography were discussed in more detail in the ‘Confidentiality’ section. One way to prove the non-repudiation of origin is to use digital signatures. The means and use of digital signatures were examined in the integrity section. Non-repudiation of delivery is harder to prove, in the case that the recipient does not sign the document received. (Dent et al., 2004)

Also secure timestamps and logs are used to ensure non-repudiation. Time-stamping is used if non-repudiation tokens are needed to have long term validity. Time stamps denote the time and date at which the certain event occurred, and the information is added into a non-repudiation receipt. Logs typically store both operations made by users and audio communication. Logs store each message with its digital signature and secure timestamp as archive records in case the non-repudiation has to be later proven. (BEA Systems)

2.2 IT Security Standards

The need for information security standards stems from the fact that customers want to make sure that the IT-products and the systems they are purchasing are secure. The other point of secure IT products is the development side. The developers want to prove that the products they are responsible for developing are secure and have been developed in the secure environment using predefined processes boosting security and quality of the products.

The IT security standards are originated from military and government fields. In commerce, the assurance that the contractors are trustworthy and the information systems are secure needs to be declared someway. Also the legal requirements and the liability of the system can be proven to be considered when a product has been properly certified. (HUT Tietoturvallisuustekniikka, 2008)

Nowadays there are three main ways when the customers want to make sure a product is developed in a secure manner. The first is that a customer knows the supplier they are working with and trusts its expertise in developing a secure IT product. The second way is for a customer to perform their own audit. Usually in this case the standards are country specific, for example, the BSI (Bundesamt für Sicherheit in der Informationstechnik) certificate in Germany. The third way of assuring the secureness of the product is to require an international security standard used in a product. (Rinne, 2010)

The IT security standards can be categorized into two groups: the standards for products and processes, and the professional certificates for experts on security. In this thesis we concentrate only on the first mentioned standards, but a short overview of people certificates is provided here. Professional security certificates are like educational degrees but more specific. As an example, the CISSP (Certified Information Systems Security Professional) certificate includes training, exams and membership of a professional society to gain enough knowledge about information security management. SANS CIAG (System Administration, Networking and Security Institute's Global Information Assurance Certification) is a more practical network security oriented and a technical certification for a specified area (such as firewall security or intrusion detection). (TKK Tietoturvallisuustekniikka, 2008)

As already stated, IT security standards for products can be divided into two categories: country specific and international standards. In Finland, the Confederation of Finnish Industries EK (Elinkeinoelämän keskusliitto), the Ministry of Interior (Sisäasiainministeriö) and the Ministry of Defense (Puolustusministeriö) have created a National Security Auditing Criterion in order to unify security audits performed in a company by a state authority. The criterion is meant also to help companies and other organizations in their internal security work. There are four main areas in this criterion from which one is IT security. Other areas are security management, personnel security and physical security. The IT security criterion is divided into seven divisions: administrative, personnel, physical, telecommunication, data system, and database and usage security. There are four levels for each division question: starting level recommendations, basic level requirements, elevated level requirements and high level requirements. As an example, the question I 501.0 asks whether the users are identified and authenticated before accessing the organization's data network and data systems. The starting level recommendation is that this is done, and the requirements grow to personal IDs and strong user IDs as the level advances. (Kansallinen Turvallisuusauditointikriteeristö, 2009)

The Federal Information Processing Standard, FIPS 140-2, is a U.S government computer security standard for software and hardware cryptographic modules. The standard from the American National Institute of Standards and Technology sets requirements for cryptographic modules which are tested by the accredited laboratories. The security requirements cover 11 areas related to the design and implementation of a cryptographic module. There are 4 levels of security rating for each area of a module, named from 1-4 (low to the highest level of security). (NIST, 2010)

From the FIPS 140-2 standard, publicised for the first time in 2001, the ISO has derived a set of international security standards. The ISO19790 standard, the Security requirements for cryptographic modules, is derived from NIST FIPS 140-2 (2001) and the ISO 24759 standard, Test requirements for cryptographic modules, specifies the methods to be used by testing laboratories to test whether a cryptographic module conforms to the requirements specified in ISO/IEC 19790. (ISO, 2010)

The international standard Systems Security Engineering – Capability Maturity Model, SSE-CMM, developed in the 1990s concentrates more on an organization’s security engineering process than the product itself. SSE-CMM’s intention is to ensure good security engineering by offering a tool for engineering organizations to evaluate their security engineering practices throughout the whole life cycle, and define improvements to the practices. In this way the developed products are guaranteed to be secure as well. The SSE-CMM appraisals can be done for organization or project level. There are 22 process areas that are evaluated with a capability level from one to five. (SSE-CMM, 2006)

Although the Trusted Computer System Evaluation Criteria, TCSEC, is no longer in use, it was the first IT security standard, and has been used as a forerunner for the models which have been developed since. TCSEC was developed in the beginning of the 1980s by the US government. The standard is commonly known as Orange Book, based on the colour of the standard’s cover (as the number of computer security standards by the US government, they became known as the rainbow series). TCSEC was the first one to set six different evaluation classes C1 being the lowest, followed by C2, B1, B2, B3 and A1 which is the highest level. TCSEC sets both functional requirements for the finished product as well as assurance requirements mostly for the development process. (TKK Tietoturvallisuustekniikka, 2008)

Common Criteria IT security standard in product development process

An international IT security standard called Common Criteria is examined in this thesis. Common Criteria has evolved directly from the Orange Book. The Common Criteria standard is discussed in detail in the following chapters.

3. Common Criteria for Information Technology Security Evaluation

This chapter consists of five sections. Introduction section discusses the purpose and the history of the Common Criteria (CC). Second section Components of the Common Criteria standard explains the terminology used by CC, the main parts of the CC standard as well as the Common Criteria stakeholders. Section 3.3 studies the notation of Common Criteria as well as the CC part 2 and 3 main components. The Common Criteria's seven evaluation assurance levels are examined in section 3.4 and the differences between evaluation, certification and accreditation and their processes are discussed in the fifth section for IT security evaluation.

3.1 Introduction

The Common Criteria for Information Technology Security Evaluation is an international standard for IT technology. The purpose of CC is to allow users to specify their security requirements, developers to specify the security attributes of the products and evaluators to determine whether the products actually meet their claims. The Common Criteria also presents the requirements for the IT security of a product and the process of implementing the security features into the product. (Mellano et al, 2007)

The Common Criteria standard is considered today as "the" international standard of IT security. In February 2010, 26 countries including Finland signed the mutual recognition agreement which means that products certified in one country are recognized in another. To demonstrate the internationality of the recognition of the standard some examples of member countries are USA, Australia, Germany, France, Sweden, Japan and India. (Common Criteria Portal, 2010)

The Common Criteria was approved as an international standard by the International Organization for Standardization (ISO) as receiving ISO/IEC 15408 for the first time in 1999. However, the foundation of the CC standard dates back to 1970s. One of the first COMPUSEC standards, DoD 5200.28-M (Techniques and Procedures for Implementing, Deactivating, Testing and Evaluating Secure Resource-Sharing ADP Systems) stated in January 1973 that the security testing and evaluation procedures will be published following additional testing and coordination. From there, as the progenitor of the Common Criteria, the Trusted Computer System Evaluation Criteria (TCSEC) standard answered the previous promises. The TCSEC standard from the United States, commonly known as Orange Book, was published in 1983, and a second version was issued in 1985. In 1991, a European security standard, the Information Technology Security Evaluation Criteria (ITSEC), was developed by France, Germany, the United Kingdom and the Netherlands. Around the same time a Canadian security standard, the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), was published in 1993. From these three standards the Common Criteria has been developed. Figure 6 presents the predecessor organizations of Common Criteria. (Herrmann, 2003)

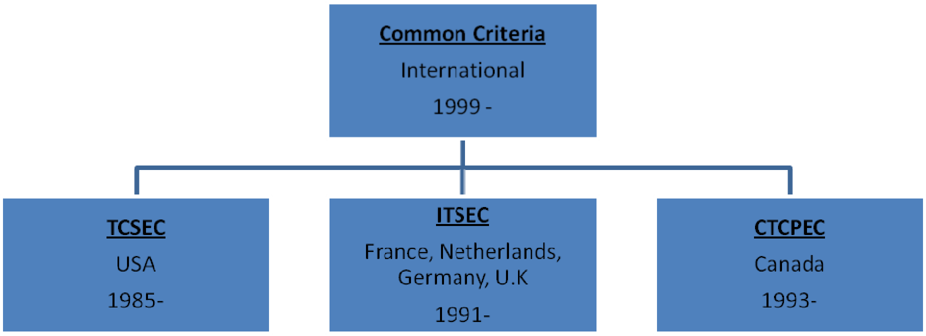


Figure 6. Predecessors of the Common Criteria

In 1993, the organization countries of TCSEC, ITSEC and CTCPEC decided to pool their resources to meet the evolving security challenges arising from the rapid changes in technology and the more universal use of information technology. The project became known as Common Criteria. In 1996, the first committee draft of CC was launched for the public to comment and to review. Three years later the Common Criteria version 2.1 was issued as the international standard ISO/IEC 15408 (Herrmann, 2003). In February 2010, the valid version available in the CC Portal of the Common

Criteria is version 3.1, a revision of version 3 dating from July 2009 (Common Criteria Portal, 2010).

In addition to the written IT security standard, the annual International Common Criteria Conference (ICCC) has been held since the year 2000. According to the official Common Criteria Portal, ICCC is the main marketing and meeting opportunity for all those involved in the specification, development, evaluation and validation or certification of IT security. The event brings together different stakeholders from certification bodies, evaluation laboratories, experts, policy makers, to product developers. (Common Criteria Portal, 2010)

3.2 Components of the Common Criteria Standard

The Common Criteria standard consists of three parts: Introduction and general models, Security functional components and Security assurance components. In addition, there is a part for evaluators called Common Methodology for Information Technology Security Evaluation (CEM). According to the Common Criteria portal (2009), the Common Criteria with CEM are the technical basis for the international agreement which assures that products can be evaluated by independent licensed laboratories, also that the Common Criteria certification process is supported by defined documents, and that the certificates are recognized by the countries signed the recognition agreement.

The Introduction part of CC describes the terms and definitions used in the Common Criteria methodology and gives an overview of the IT standard in terms of, for instance, the evaluated product and target audience of CC. Furthermore, the introduction part also represents the general model of CC and explains how the tailoring of security requirements can be applied. The first part also describes what protection profiles and packages are and how an evaluation process defines the results. The second CC part, ‘Security functional components’, describes the desired security behaviour of the evaluated product whereas the third CC part, ‘Security assurance components’, defines the evaluation criteria and presents seven different evaluation assurance levels. Part 2 and part 3 are examined more in details in the following sections. CEM, mainly targeted towards evaluators, describes the evaluation process and related tasks and describes the evaluation criteria for each class. In addition to the actual standard material, CC has, for example, also published guides for developer documentation and for transition from the older CC version 2.3 to the new CC version 3.1 (Common Criteria, 2009)

Common Criteria uses its own terminology which is good to understand before going into the details of the content of the standard. Table 1 summarizes the most important Common Criteria terminologies and abbreviations with short explanations.

Table 1. The Common Criteria terminology

Abbreviation	Term	Explanation
CC	Common Criteria	The name of the IT standard
TOE	Target of Evaluation	Product or system to be evaluated
TSF	TOE Security Functionality	Part of the evaluated product where the security is implemented
TSFI	TSF Interface	The interfaces used by users to interact with security functions
PP	Protection Profile	Document describing standard security requirements for a generic type of product
ST	Security Target	Main document specifying TOE and evaluation tasks
SFR	Security Functional Requirement	Common Criteria part 2 describing functional components
SAR	Security Assurance Requirement	Common Criteria part 3 describing assurance components
EAL	Evaluation Assurance Level	CC part 3, level of assuring the evaluation

The following definitions are directly according to the Common Criteria standard (2009) whereas the explanations of the meanings in practice are from Oppida training material (2010).

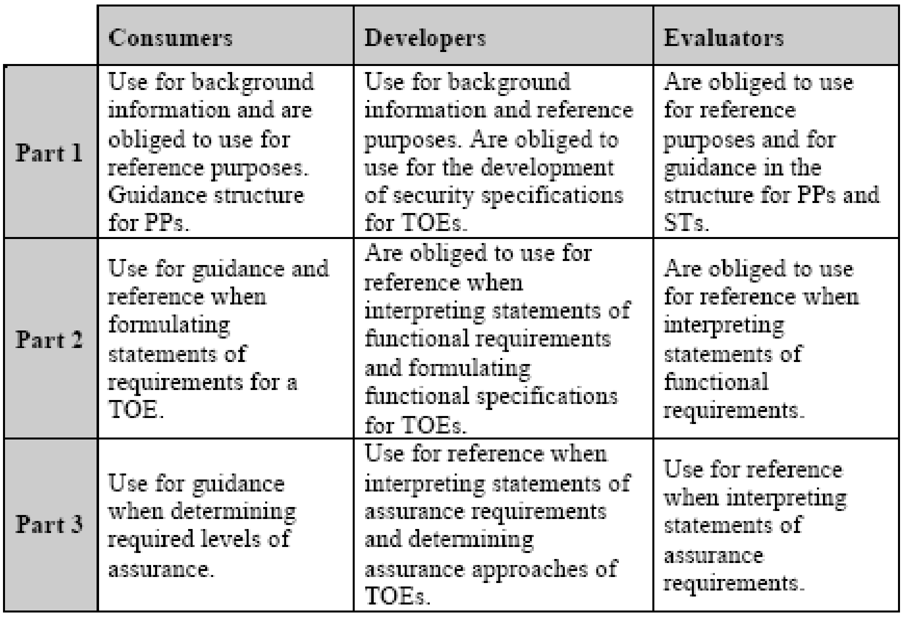
The target of evaluation, TOE, is defined as a "set of software, firmware and/or hardware possibly accompanied by guidance". In short, the TOE is the product chosen to be evaluated. The TOE Security Functionality, the TSF, is defined as the "combined functionality of all hardware, software and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs". In practice this means that TSFs are the parts of the evaluated product where the security is implemented. The TSF Interface (TSFI) is defined according to CC as "means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF". TSFIs are the interfaces used by users to interact with security functions. The Protection Profile, PP, is "an implementation-independent statement of security needs for a TOE type". That is, a document specifying for a generic product (such as a smart card) a standard set of security requirements. Compared to the definition of Protection Profile, the Security Target, ST, is an "implementation-dependent statement of security needs for a specific identified TOE". A Security Target document specifies the security requirements to a defined product and its evaluation tasks. The Security Functional Requirement, SFR, describes the "desired security behaviour expected of a TOE … and properties that users can detect by direct interaction with the IT or by the IT response to stimulus". SFRs form the CC part 2 listing different possible functional components in ways that a product can fulfill within its security features and counter threats. Security Assurance Requirements, SARs, "establish a standard way of expressing the assurance requirements for TOEs". SARs form the CC part 3 by cataloguing a set of assurance components that build up the Evaluation Assurance Levels, EALs. An EAL is a "set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale that form an assurance package". In other words, EALs define a set of requirements for assuring the security depending on the level.

In order to understand the overall view of the methodology, a simplified description is given. When starting to plan and develop a product to fulfill the requirements of the CC standard, the evaluated product, TOE, has to be specified with its security features, TSFs. In order to understand the functionality and possible security threats for the product, the interfaces of security functions, TSFIs, are also specified. If the generic product has a defined Protection Profile document or a customer does the PP, it is used as the basis for constructing a Security Target, ST; the document describing the chosen specifications of the TOE, TSFs and evaluation tasks. Security Functional Requirements, SFRs, form the basis for stating the security functions of the TOE whereas Security Assurance Requirements, SARs, affect the development process based on the chosen Evaluation Assurance Level, EAL.

There are three generic groups of users with an interest in the CC standard: consumers, developers and evaluators. The fundamental purpose of the Common Criteria is to fulfill the needs of the consumers. Consumers can use the evaluation results to compare different products and to identify whether a product fulfills their security needs. Consumers can also express their security requirements in an unambiguous way by creating a Protection Profile of a generic product. (CC part 1, 2009)

Developers are the organizations and individuals who design, build and sell IT security products. Developers specify the product specific security requirements in the Security Target which may respond to a consumer's Protection Profile (Herrmann, 2003). CC is also intended to support developers for preparations in the evaluation of their TOE. Evaluators perform evaluations and form judgements about the conformance of TOE according to the general actions described by CC. Table 2 represents the three key target audience groups and describe how the parts of the CC are used by each group. The table is copied directly from CC part 1 "Road map to the Common Criteria". (CC part 1, 2009)

Table 2. CC stakeholders and their use of the standard's parts (CC part 1, 2009)



	Consumers	Developers	Evaluators
Part 1	Use for background information and are obliged to use for reference purposes. Guidance structure for PPs.	Use for background information and reference purposes. Are obliged to use for the development of security specifications for TOEs.	Are obliged to use for reference purposes and for guidance in the structure for PPs and STs.
Part 2	Use for guidance and reference when formulating statements of requirements for a TOE.	Are obliged to use for reference when interpreting statements of functional requirements and formulating functional specifications for TOEs.	Are obliged to use for reference when interpreting statements of functional requirements.
Part 3	Use for guidance when determining required levels of assurance.	Use for reference when interpreting statements of assurance requirements and determining assurance approaches of TOEs.	Use for reference when interpreting statements of assurance requirements.

3.3 Security Functional and Assurance Requirements

3.3.1 Notation

The Common Criteria Security Functional Requirement, part 2, and Security Assurance Requirements, part 3, are so-called catalogues where different security requirements are divided into classes. The requirements are systematized to have a hierarchical structure with a standard notation of ABB_CCC.x.yy.

The first letter of the notation "A" is either F or A portraying either the Functional or Assurance component of the CC (part 2 or 3). The next two letters "BB" define the class. The classes of SFRs and SARs are explained more in detail in this section. The "CCC" defines the family code with three letters. Each class has at least one, but in most cases, several families. The functional family states its security objective and general description of the functional or assurance requirements. Families are divided into 1-digit components "x". Assurance requirement components are leveled in terms of scope, depth and/or rigor, for functional requirements this is only in some times the case. A component can have several elements "yy". In the functional requirements, only the first digit is used stating the requirement number. In the assurance requirements, the first "y" describes the serial number of the requirement and the second digit describes whether the action is meant for a developer (D) or for an evaluator (E) or whether it describes the content (C) and presentation of the requirement.

As an example, the AGD_OPE.1.1D "The developer shall provide operational user guide", the notation shows that this requirement belongs to the Assurance class called Guidance documents (AGD). The family, OPE, comes from Operational user guidance. The component and element 1.1D shows that this is the first component and first element action for the developer stating what the developer is expected to do. Figure 7 presents the standard notation of CC with an example of Functional class of Audit (FAU), family of Security audit automatic response (ARP) with its first component and element (1.1). (CC part 2, part 3, 2009)

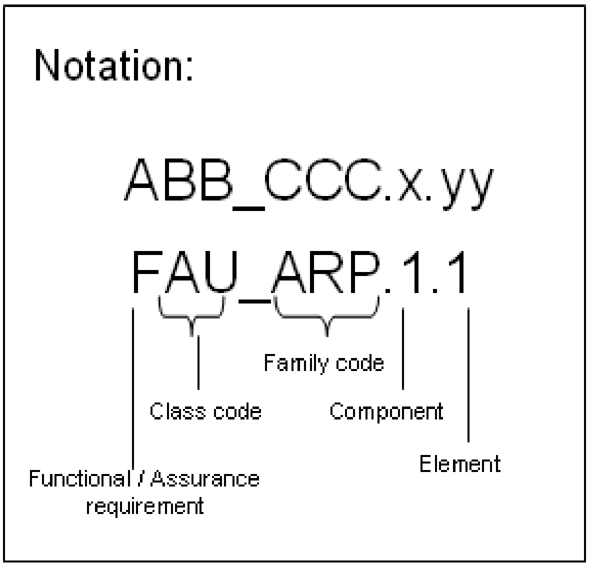


Figure 7. Common Criteria Standard Notation

3.3.2 Security Functional Components

The Security Functional Components, part 2 of CC, describes the widest possible spectrum of security functions that a consumer may need in an IT product. The most suitable functions from the catalogue are chosen and listed in the Security Target document (created first in the process). The writer of Protection Profile and Security Target documents must be familiar with the Security Functional Components and their families.

Security Functional Components consist of eleven classes. The classes have no hierarchical relationship with other classes but the classes are alternative to one another. This is the reason why the CC presents the classes in alphabetical order by the class code. As already stated, a consumer or a developer can select the security requirements from the classes that the product will include. One IT product is not expected to contain security requirements from all of the classes but have a selective variety of security components. (Herrmann, 2003)

The eleven classes are shortly represented in terms of their respective security focus area in order to give an overview of the CC part 2 and possible security features that an IT product can have. Table 3 summarizes the functional classes with their abbreviations.

Table 3. Functional classes

Class	Abbr.	Description
Security Audit	FAU	information and security related activities that determine what has happened and by whom
Communication	FCO	assurance of identity of participating parties
Cryptographic support	FCS	encryption key management and operations
User data protection	FDP	requirements for protecting user data
Identification and Authentication	FIA	establishment and verification of claimed user identity
Security Management	FMT	management of TOE's security functions
Privacy	FPR	privacy to provide user protection
Protection of the TSF	FPT	protection of TOE security functions and its data
Resource utilization	FRU	ensuring the availability of required resources
TOE Access	FTA	controlling the establishment of a user's session
Trusted path / channels	FTP	ensuring a trusted communication path between TSF and the user or other IT product

Security Audit (FAU)

According to CC part 2 v3.1 rev3 (2009), Security auditing involves recognizing, recording, storing, and analyzing information related to security relevant activities (i.e. activities controlled by the TSF). The resulting audit records can be examined to determine which security relevant activities took place and which user is responsible for them. Security Audit families support both traditional logging, storing and reporting as well as detecting actual and potential security violations. The Security Audit class consists of six families, for instance Security Audit Analysis and Review.

Communication (FCO)

Communication class describes requirements to assure the identity of a party participating in a data exchange. There are two families in the communication class. The first family deals with non-repudiation of origin: to assure the identity of the originator of the transmitted information. The second family is to ensure the non-repudiation of receipt: verifying the identity of the recipient of the transmitted information. (CC part 2, 2009)

Cryptographic Support (FCS)

The FCS class is taken into account when the TOE uses encryption either in hardware, firmware and/or software. Cryptographic support consists of two families, cryptographic key management and cryptographic operation. The first named address the management aspects of cryptographic keys, and the second discusses the operational use of the cryptographic keys. (CC part 2, 2009)

CC does not state which encryption algorithms are acceptable. Instead it concentrates on the secure use of encryption by the TOE. (Herrmann, 2003)

User Data Protection (FDP)

User data protection specifies requirements related to protecting user data. The class consists of 13 families which are split into four groups. The first group addresses user data protection security function policies such as access control and information flow control policies. The second family is about forms of user data protection. The third group addresses the requirements for trustworthy transfer into or out of the product. The fourth group of families addresses the inter-TSF communication: the communication between the security functions of the product and another IT product. (CC part 2, 2009)

Identification and Authentication (FIA)

The FIA class addresses the requirements for functions to establish and verify (to identify and to authenticate) a claimed user identity. The purpose of this class is to ensure that users have proper security attributes such as identity, groups and security levels determined and verified. The FIA class consists of six families. The FIA has effects also on other classes, for example on User Data Protection and Security Audit. (CC part 2, 2009)

Security Management (FMT)

Security Management specifies requirements for managing the product's security functions and their security attributes and data. Also the different management roles and their interaction (e.g. separation of capability) can be specified. FMT consists of 7 families such as management of TSF data and revocation. (CC part 2, 2009)

Privacy (FPR)

The FPR class contains privacy requirements that provide user protection against discovery and misuse of identity by other users. The Privacy class contains four families: anonymity, pseudonymity, unlinkability and unobservability. (CC part 2, 2009)

Protection of the TSF (FPT)

Protection of the TSF class contains requirements for protecting TOE security functions and TOE security function data. There are three significant elements in FPT: execution and implementation of the mechanisms that enforce the SFRs, the administrative databases that guide the enforcement of the SFRs, and the external entities that the TSF may interact with. The FPT class is divided into 14 families such as availability of exported TSF data and TSF physical protection. (CC part 2, 2009)

Resource Utilization (FRU)

The Resource of Utilization class ensures the availability of required resources such as processing and storage capability. FRU consists of three families. Fault tolerance provides protection against unavailability of capabilities caused by failure of the TOE. The family Priority of Service ensures that the resources will be allocated to the more important tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolizing the resources. (CC part 2, 2009)

TOE Access (FTA)

The Target of Evaluation Access class specifies functional requirements for controlling the establishment of a user's session. FTA is composed of six families which ensure, for instance, the limitation of concurrent multiple sessions, the requirements for session locking and termination, and TOE access history. (CC part 2, 2009)

Trusted Path / Channels (FTP)

The FTP class provides requirements for a trusted communication path between the users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. A trusted path provides users to perform functions through an assured direct interaction with the TSF. A trusted channel is a communication channel that can be initiated by either side of the channel, and provides non-repudiation characteristics in order to identify the sides of the channel. FTP consists of an inter-TSF trusted channel and trusted path families. (CC part 2, 2009)

3.3.3 Security Assurance Components

Security Assurance Components, part 3 of CC, describe requirements for the development process. Depending on the selected evaluation assurance level, the requirements alter. While the Security Assurance Components cover the whole process, all the developers should be familiar with the components and especially know the parts that they are responsible for.

The Security Assurance Component consists of eight classes (CC part 3, 2009). The assurance classes have hierarchical relationship with other classes and for that reason they are presented by the order of the creation process. Security assurance requirements are invoked to ensure that all security functional requirements, the IT environment and the non-IT environment, have been implemented correctly and that they are sufficiently robust to counter identified threats. (Herrmann, 2003)

It is good to notice that the assurance classes have been altered in different versions of the Common Criteria and that’s why the older Common Criteria source materials and evaluated products have somewhat different classes than presented in this thesis. The eight classes are shortly represented in terms of its assurance focus area. Table 4 summarizes the assurance classes with their abbreviations.

Table 4. CC Assurance Classes

Class	Abbreviation	Explanation
Protection Profile evaluation	APE	evidence that a Protection Profile is sound and consistent and can be used as a basis for ST
Security Target evaluation	ASE	evidence that a ST is sound and consistent and can be used as a basis for TOE evaluation
Development	ADV	information about the TOE and TSFs
Guidance Documents	AGD	guidance documents for secure handling of the TOE
Life-cycle Support	ALC	development and maintenance procedures of the TOE
Tests	ATE	evidence that the TSFs behave as described
Vulnerability Assessment	AVA	address the possibility of exploitable vulnerabilities
Composition	ACO	assurance that a composited product from previously evaluated TOEs operates securely

Protection Profile Evaluation (APE)

The purpose of the Protection Profile evaluation class is to set requirements which have to be met in a Protection Profile (PP) document. The overall goal is that PP is sound and internally consistent such a way that a PP can be used as the bases for writing a Security Target document or another Protection Profile. If a PP is based on other PPs, the PP has to show that it has used the information correctly. The six families of the Protection Profile evaluation specify the structure and the content of a PP, for example stating the PP introduction and security objectives chapters. The annexes of part 1 of the Common Criteria also give more specific information and examples of how a PP can be constructed. (CC part 3, 2009)

Security Target Evaluation (ASE)

The purpose of the Security Target evaluation class is to set the requirements which have to be met in a Security Target (ST) document. Like PP, evaluating a Security Target is required to show that the ST is sound and internally consistent in such a way that the ST can be used as a basis for a TOE evaluation. If a ST is based on one or more PPs, the ST has to show the correct use of these PPs. The seven families of ASE describe the structure and the content of a ST in the same way as in APE. The Common Criteria part 1 annexes also give more specific information and examples how a ST can be constructed. (CC part 3, 2009)

Development (ADV)

The development class provides the requirements regarding the information about the Target of Evaluation (TOE). The ADV class is composed of six families that structure and represent the TOE Security Functionality (TSF) at various levels. The purpose of documenting the requirements of the ADV class is to show that the security functionality is performed and specified, and that the TOE cannot be used in a way that the security functionality can be corrupted or bypassed. The information from the ADV class is used as a basis for conducting vulnerability analysis (AVA class) and testing of the TOE (ATE class). (CC part 3, 2009)

Guidance Documents (AGD)

The Guidance Documents class provides the requirements for guidance documentation for all user roles. Guidance documents should describe aspects for the secure handling of the TOE. AGD class also sets requirements for addressing the possibility of unintended incorrect configuration or handling of the TOE. The AGD class is divided into two families: preparative procedures and operational user guidance. (CC part 3, 2009)

Life-cycle Support (ALC)

The Life-cycle Support class sets the requirements for discipline and control in the development and maintenance processes of the TOE. The ALC class documents should also state the point where the TOE is handed over to the user's responsibility.

The ALC class consists of seven families that define, for example, the TOE life-cycle, configuration management, security of development and delivery procedures. (CC part 3, 2009)

Tests (ATE)

The purpose of the Tests class is to assure that the TOE Security Functionality (TSF) behaves as described in the design descriptions from the development class. The ATE class is composed of four families that set the requirements for functional tests, testing coverage, the depth of testing and independent testing (i.e. evaluator testing). (CC part 3, 2009)

Vulnerability Assessment (AVA)

The purpose of the Vulnerability Assessment class is to address the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE. The AVA class evaluates potential vulnerabilities that could allow attackers to violate the security functional requirements by unauthorized access to data and functionality, or inferring and altering the TSF or authorized capabilities of other users. The AVA class consists of only one family, vulnerability analysis that the evaluators perform. (CC part 3, 2009)

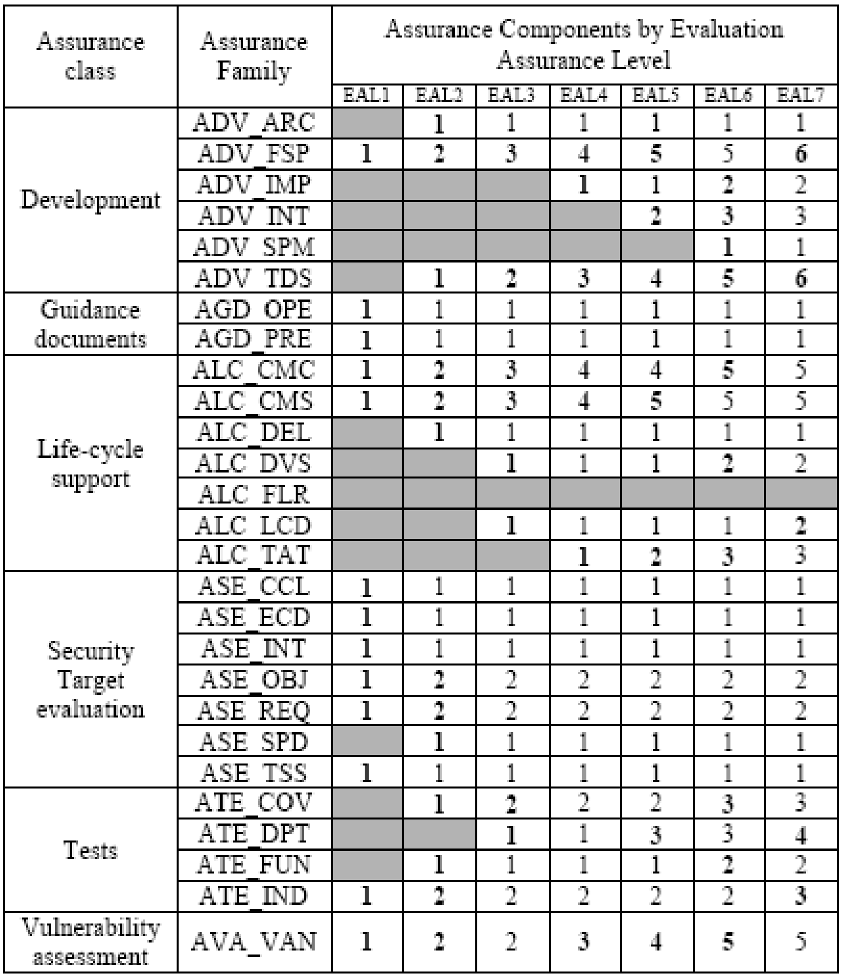
Composition (ACO)

The composition class is used in cases where two or more CC evaluated products are combined into use without further development of the evaluated IT products. The ACO class specifies the assurance requirements that are designed to provide that a composed TOE will operate securely when relying upon security functionality provided by a previously evaluated product. The ACO class consists of five families such as composition rationale, development evidence and composed TOE testing. (CC part 3, 2009)

3.4 Common Criteria Evaluation Assurance Levels

Common Criteria part 3 (2009) defines seven evaluation assurance levels (EALs), EAL1 being the lowest and EAL7 the highest assurance level. EALs balance the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance. Evaluation assurance levels are composed of security assurance requirements (SARs). The increase in assurance level is made by substituting a hierarchically higher assurance component from the same assurance family (i.e. increasing rigour, scope and/or depth) and by adding new requirements from other families. Also variations of combinations of assurances are possible, and by adding an assurance component or components to a certain level, the notion of "augmentation" such as EAL3+ can be achieved. However, it is important to notice that the EAL, as the name states, tells only the assured level of evaluation and it is possible that in reality a product with a lower EAL can actually be more secure than some other product with a higher EAL. Table 5 represents a summary of the EALs directly from CC part 3 (2009). The columns represent a hierarchically ordered set of EALs and the rows represent assurance families. Each number in the resulting matrix identifies a specific assurance component where applicable. (CC part 3, 2009)

Table 5. Evaluation Assurance Levels



Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		1	1	1	1	1	1
	ADV_FSP	1	2	3	4	5	5	6
	ADV_IMP				1	1	2	2
	ADV_INT					2	3	3
	ADV_SPM						1	1
	ADV_TDS		1	2	3	4	5	6
Guidance documents	AGD_OPE	1	1	1	1	1	1	1
	AGD_PRE	1	1	1	1	1	1	1
Life-cycle support	ALC_CMC	1	2	3	4	4	5	5
	ALC_CMS	1	2	3	4	5	5	5
	ALC_DEL		1	1	1	1	1	1
	ALC_DVS			1	1	1	2	2
	ALC_FLR							
	ALC_LCD			1	1	1	1	2
	ALC_TAT				1	2	3	3
Security Target evaluation	ASE_CCL	1	1	1	1	1	1	1
	ASE_ECD	1	1	1	1	1	1	1
	ASE_INT	1	1	1	1	1	1	1
	ASE_OBJ	1	2	2	2	2	2	2
	ASE_REQ	1	2	2	2	2	2	2
	ASE_SPD		1	1	1	1	1	1
	ASE_TSS	1	1	1	1	1	1	1
Tests	ATE_COV		1	2	2	2	3	3
	ATE_DPT			1	1	3	3	4
	ATE_FUN		1	1	1	1	2	2
	ATE_IND	1	2	2	2	2	2	3
Vulnerability assessment	AVA_VAN	1	2	2	3	4	5	5

As seen from Table 5, some of the assurance families have the same requirement for every assurance level. The number "1" in the whole row, for example, for Guidance documents and most of the Security Target evaluation families describes that the requirements in every level are the same. The classes where the level required adds the most competence are in the development classes, configuration families (ALC_CMC, ALC_CMS) and in the vulnerability assessment class.

The seven evaluation assurance levels are briefly discussed to give an overview of the assurance requirements of each level.

EAL1 – functionally tested

EAL1 provides a basic level of assurance. EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. EAL1 is the only level that a limited Security Target is acceptable, and the evaluation is made by testing against a specification and an examination of the guidance documents provided to understand the security behaviour. (CC part 3, 2009)

EAL1 is appropriate for products meeting specific customer needs requesting low assurance (CC Developers Guide, 2009).

EAL2 – structurally tested

EAL2 provides a low to moderate level of independently assured security. EAL2 requires from the developer the delivery of design information and test results but shouldn't require a substantially increased investment of cost or time. EAL2 requires increase in assurance from EAL1 by adding developer testing, a vulnerability analysis and independent testing based upon more detailed TOE specification. (CC part 3, 2009)

EAL2 is a level which is often used for applications with security functionality. It is also an “entry level” for developers who are aiming at a higher level but want first to become familiar with the evaluation and certification process. (CC Developers Guide, 2009)

EAL3 – methodically tested and checked

EAL3 is suitable for those situations where a moderate level of independently assured security and a thorough investigation of the TOE and its development are wanted. However, substantial alteration of existing development practices is not needed. Compared to EAL2, EAL3 has a meaningful increase in assurance by requiring more complete testing coverage of the security functionality and mechanisms and/or procedures that provide some confidence that the product will not be tampered with during the development. (CC part 3, 2009)

EAL3 is a level which is typically selected for complex products (like operating systems) in the case that a higher level of security is considered to be too costly (CC Developers Guide, 2009).

EAL4 –methodically designed, tested, and reviewed

EAL4 is applicable where a moderate to high level of independently assured security is demanded and additional security-specific engineering costs are accepted. EAL4 is the highest level at which it is still in normal cases used to retrofit an existing product line. EAL4 requires more design descriptions, the implementation representation for the entire TSF, and improved procedures that provide confidence that the TOE will not be tampered with during development compared to EAL3. (CC part 3, 2009)

EAL4 is a level which is adequate for products and customers mandating requiring a high assurance level such as firewalls, smart card components (CC Developers Guide, 2009).

EAL5 – semiformally designed and tested

EAL5 provides a high level of independently assured security. The development approach is rigorous with a moderate application of specialist security engineering techniques. The product and the process are likely to be designed and developed to achieve EAL5 assurance. EAL5 adds requirements compared to EAL4 by requiring semiformal design descriptions, a more structured and analyzable architecture, improved procedures that provide confidence that the TOE will not be tampered with during development. (CC part 3, 2009)

EAL5 is a level which is selected in case customers request extra assurance (CC Developers Guide, 2009).

EAL6 – semiformally verified design and tested

EAL6 permits developers to gain high assurance from the application of security engineering techniques to a strict development environment. Therefore EAL6 is suitable for the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs. EAL6 increases the assurance level from EAL5 by requiring more comprehensive analysis, a structured representation of the implementation, more architectural structure, more comprehensive independent vulnerability analysis and improved configuration management and development environment controls. (CC part 3, 2009)

EAL6 is selected only in extraordinary cases (CC Developers Guide, 2009).

EAL7 – formally verified design and tested

EAL7 is applicable to the development of security TOEs for application in extremely high risk situations and/or where the high value of the assets justifies the higher costs. EAL7 represents complete, independent white-box testing that employs formal methods, similar to those in use by the safety engineering community. (Herrmann, 2003)

EAL7 represents an increase in assurance from EAL 6 by requiring more comprehensive analysis using formal representations and formal correspondence, and comprehensive testing (CC part 3, 2009).

EAL7 is selected only in extraordinary cases (CC Developers Guide, 2009).

The Common Criteria portal publishes statistics about the categories of evaluated products and the evaluation assurance levels. Figure 8 represents the number of evaluations made by each evaluation assurance level in 1997-2009. Most of the evaluations are done from EAL4 (488 evaluations) whereas EAL6 and EAL7 evaluations have only been made four times altogether.

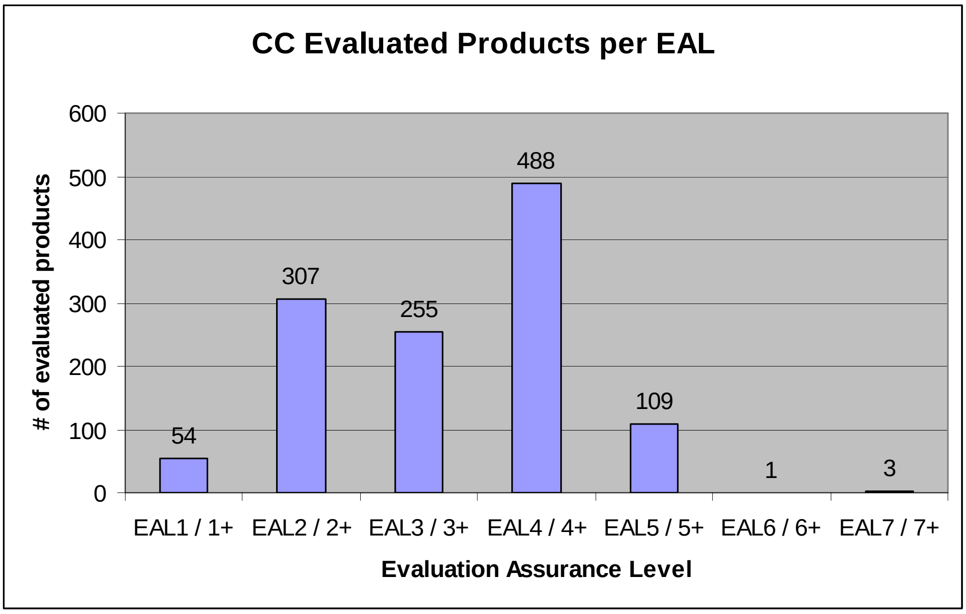


Figure 8. CC evaluated products per evaluation assurance level

3.5 Evaluation, Certification and Accrediation

There are three different levels of verifying an IT security product. The processes are called evaluation, certification and accrediation. At the moment there are no bodies in Finland who are entitled to perform evaluation, certification or accrediation. However, Finland is considered as a certificate consuming nation. This means that evaluations of Finnish products made by certified bodies are recognized internationally as well as Finland recognizes the certificates acknowledged elsewhere.

A working definition for evaluation by Anderson (2001) is "the process of assembling evidence that a system meets, or fails to meet, a prescribed assurance target. Evaluation often overlaps with testing, and is sometimes confused with it". The evaluation process according to CC defines that the evaluated product meets the security requirements mentioned in its Security Target (Oppida, 2010). Evaluation can be done by independent licensed laboratories which are listed in official Common Criteria portal (CC portal, 2010). Evaluators check the Common Criteria related documents created by developers and give intermediate reports for each activity. Evaluation is done in consecutive evaluation rounds at the same time with the product development process and the feedback is given after each evaluation "round". Evaluators state if the security assurance requirements are successfully *met*, *failed* or *inclusive* (more information is needed to give the verdict). Usually the evaluation process during the development lasts from six months to a year. (Oppida, 2010)

The number of evaluated products has increased significantly from one product in 1997 (the first year of CC) to around 200 evaluated products a year. The most evaluations made so far was in the 2007 (230 evaluated products) from where the number has decreased somewhat to 190 evaluated products in 2009. The total number of evaluations made between the years 1997 and 2009 is 1217. Figure 9 represents the number of Common Criteria evaluations made per year. (CC portal, 2010)

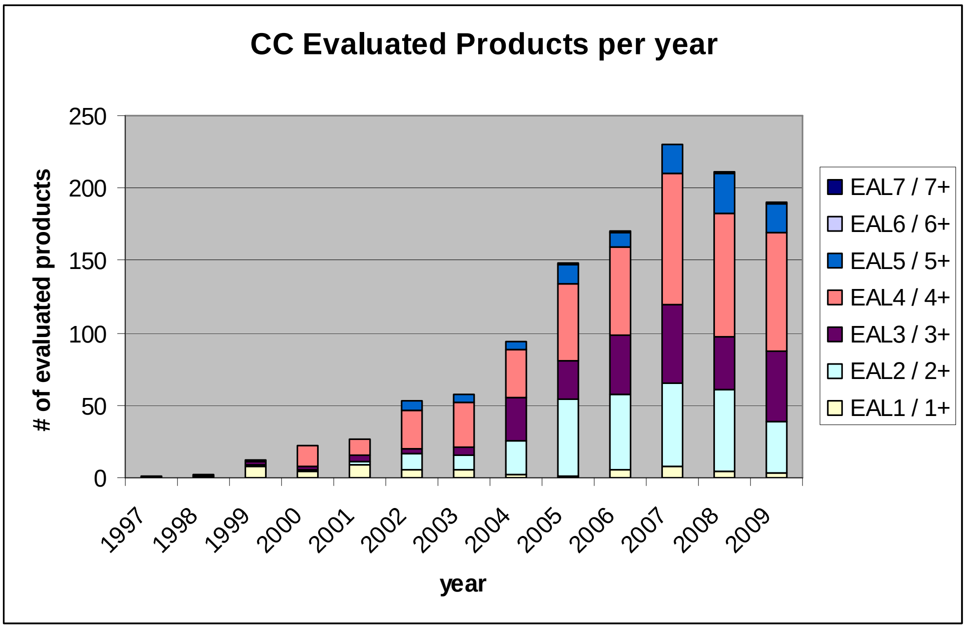


Figure 9. CC evaluated products a year

The certification process verifies that the evaluation of the product has been performed in conformity with the rules of the Certification Body in terms of independence, competence and methodology. Certification is made by the national Certification Body in the Certification Authorizing countries. The certification takes place after the evaluation process and usually takes about a month (Oppida, 2010). Countries issuing the CC certificates are Australia, New Zealand, Canada, France, Germany, Italy, Japan, Republic of Korea, the Netherlands, Norway, Spain, Sweden, the United Kingdom and the United States. Countries recognizing the CC certificates are Austria, Czech Republic, Denmark, Finland, Greece, Hungary, India, Israel, Malaysia, Pakistan, Singapore, and Turkey. (CC portal, 2010)

A system or network is said to be accredited once a formal declaration has been made by the designated approval authority that an IT system is approved to operate in a particular security mode using a predescribed set of safeguards to an acceptable level of risk (Herrmann, 2003). In other words, the accrediation process verifies that classified information can be operated in the evaluated and certified system. Accrediation of CC can be made by the Accrediation board for instance the NSAB for NATO. (Oppida, 2010)

The Common Criteria portal publishes information about certified products. The public information (mainly targeted for customers) includes the name of the product and the manufacturer, evaluation assurance level achieved, certification report and date, Security Target and possible Protection Profiles. Security Target is the main document created by the developers. The certification report is made by the evaluators, and includes mainly information about the evaluated product, each deliverable and the results of the evaluation, with possible additional comments, observations and recommendations. The layout of the certification report depends on the evaluation organization.

4. Application of CC Standard to Development Processes

This chapter reviews how the Common Criteria standard is applied in a development project using the waterfall model. Section 4.1 presents the waterfall model and its phases in an IT product development project. Section 4.2 acquaints the reader with the exact Common Criteria requirements in an evaluation assurance level 3 (EAL3) and explains how the top level schedule must be planned. Finally, section 4.3 examines each requirement family in EAL3 in terms of the deliverable.

4.1 Generic Software and Hardware Development Project Process

The case in this thesis describing how to apply the Common Criteria in a development project uses the generic waterfall model originally designed by Winston W. Royce (Chaffey et al, 2005). The waterfall model is a sequential software development process, in which the process is seen as flowing from the previous phase down to the next phase. The process starts with an analysis phase, in which the business case and user requirements for the system are investigated. After that, in the requirements specification phase the necessary features of the system are defined. Once the requirement specifications are made, the design phase creates a suitable solution how the configuration of the system can be done. When the plans and specifications are ready the implementation, the development of the proposed solution, can start. In the testing and integration phase the solution is tested to verify that the system solves the original specifications and the system works in the context. In the operation and maintenance phase the system is used by the customers, and in the case of identified problems or new requirements, the working solution may be modified. Figure 10 presents the phases of the waterfall model. (McCormack et al, 2005)

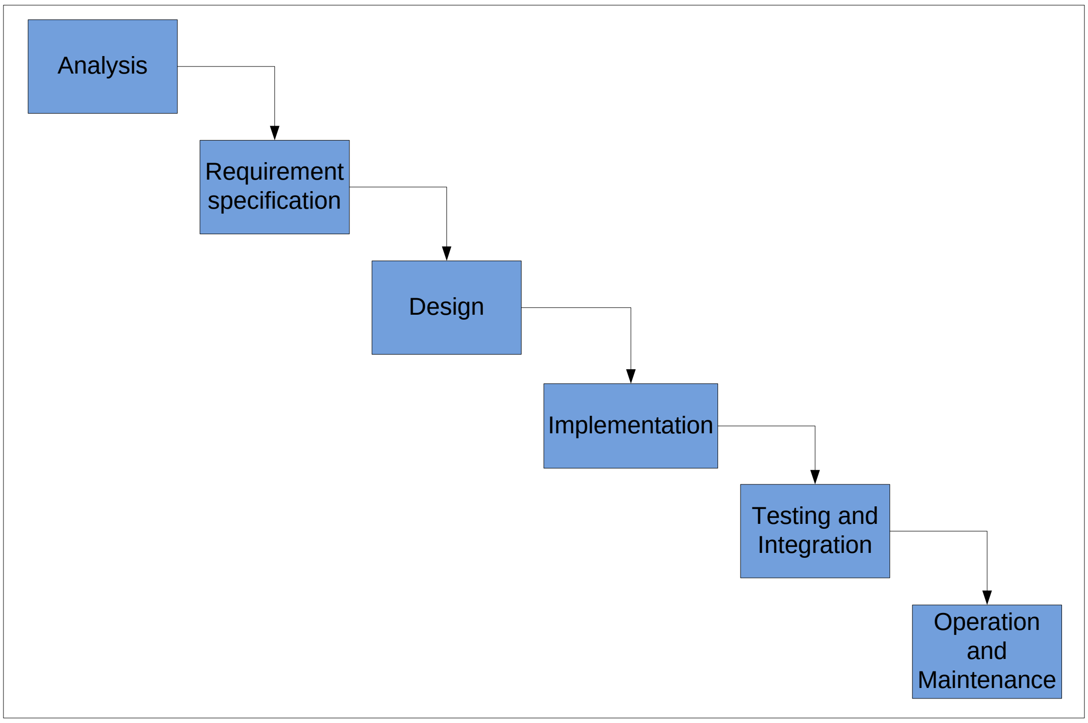
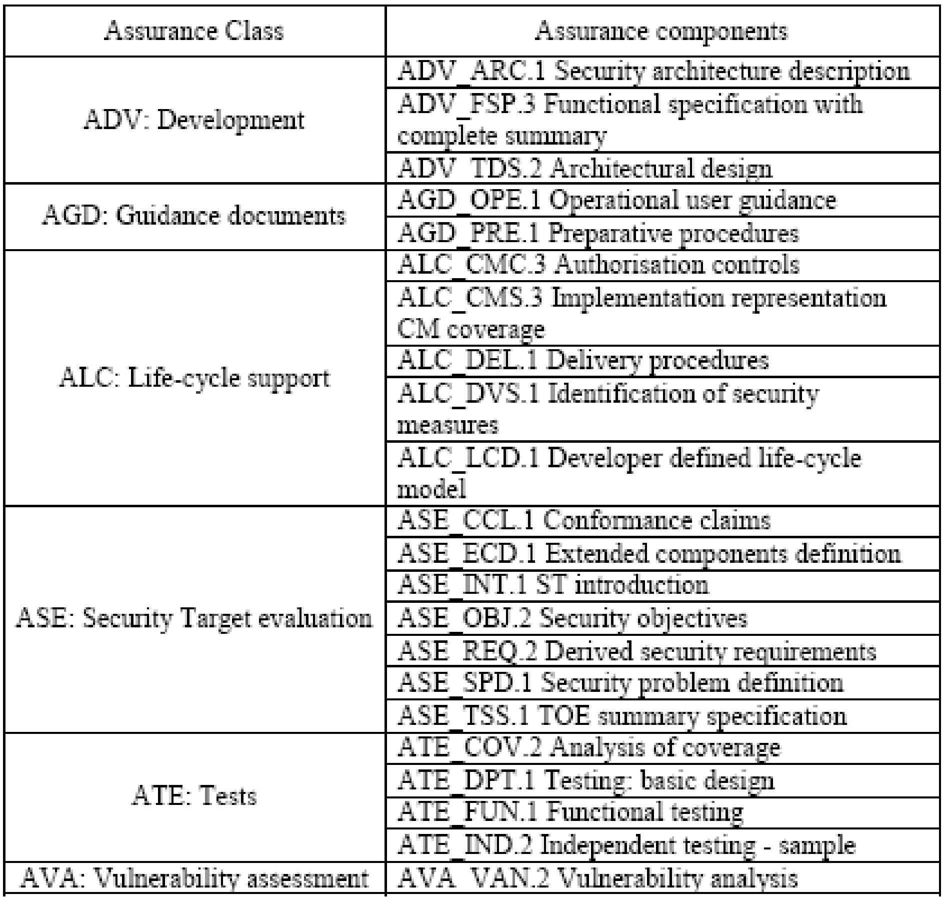


Figure 10. Waterfall model

4.2 The Common Criteria Requirements at Evaluation Assurance Level 3

The case applying CC requirements was decided to apply the CC evaluation assurance level 3 (EAL3) because of the adequate level of security assurance offered by it and because the CC requirements were used for the first time in the organization. The Common Criteria defines the assurance components to be applied by each level in CC part 3, and the assurance components for EAL3 are presented in Table 6 in an alphabetical order. The exact requirements of each component by the CC standard are listed in Appendix 1. The next section 4.3, will go deeper into each component and it is requirements in practice. Giving a short overall comparison of the components and the level of implementation to a process, level 3 does not require all the families of CC from development and life-cycle support classes. However, these are also the classes that have increased number of requirements compared to EAL2, as well as some additional tests families.

Table 6. Evaluation Assurance Level 3 components (CC part 3, 2009)



Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.2 Architectural design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.3 Authorisation controls
	ALC_CMS.3 Implementation representation CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

When applying the Common Criteria standard, the order of the generation of the deliverables is defined with a critical path. The work has to be begun with generating the Security Target, by defining the scope of the evaluated product (TOE) and describing the security features proved to be secure (TSFs). Also the life-cycle support documents describing the development model, identified security measures, configuration management procedures and delivery procedures should be defined before starting the actual development of the product. Once the ST is done, development of the design documents can start by describing architectural and interface (CC term: functional) specifications. Test plans are needed after implementation in order for the test phase to start. Once the product is being implemented and its security features are tested, documents describing test results, the installation and user guidance, and configuration management baselines are created. After that, the independent testing and vulnerability analysis by the evaluators can take place. The official Common Criteria evaluation process is done at the same time with the product development process and the evaluators evaluate the deliverables once they are being produced. The evaluation of each deliverable will take time and this has to be taken into account when planning the time schedule of the project and possible customer commitments. Usually during the the evaluation, several versions of each deliverable are needed in order to fulfill the CC requirements. Figure 11 by Oppida (2010) shows the order of the deliverables produced and the approximate time that the evaluators will need for each class of requirements. As seen from the figure, the time to evaluate the product at EAL3 will take at least 7 months.

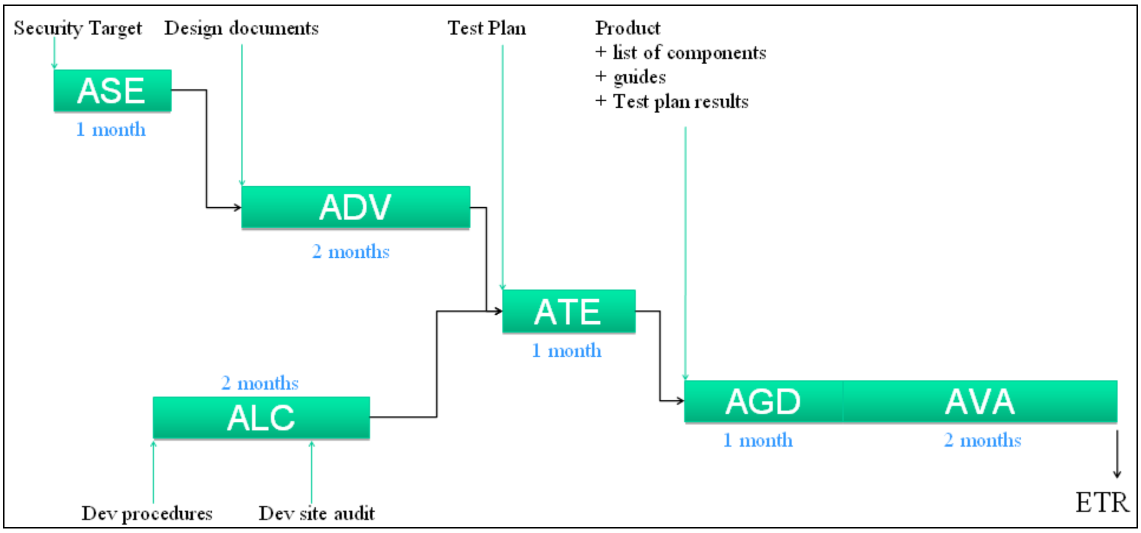
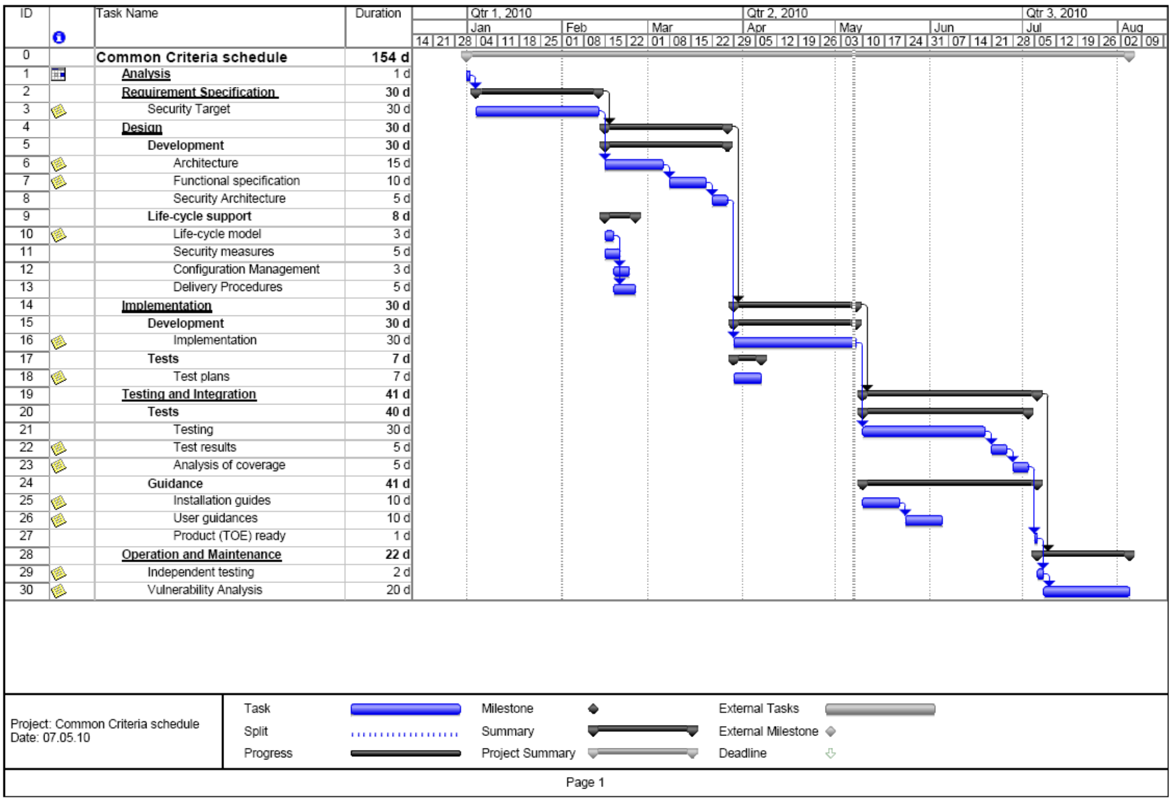


Figure 11. Schedule of the deliverables for evaluation (Oppida, 2010)

When combining the Common Criteria requirements and the evaluation process with the project's waterfall model process, a high level plan of each deliverable to be produced in the right phase is presented in Figure 12. The bolded and underlined headers are the phases in the waterfall model. Under each phase the right classes of Common Criteria with its deliverables are described. The duration of each deliverable is estimated only to give an overview of the workload of the task. The actual work needed for each task depends on the scope of the TOE. The project is set to start in the beginning of the year in order easily to see the total needed time for producing the Common Criteria requirements. The time needed for evaluation is not taken into account in the figure. The arrows in the picture link the task which affects the following task. A larger picture of Figure 12 is attached as Appendix 2.



ID	ℹ	Task Name	Duration
0		**Common Criteria schedule**	**154 d**
1	▦	**Analysis**	1 d
2		**Requirement Specification**	**30 d**
3	📝	Security Target	30 d
4		**Design**	**30 d**
5		**Development**	**30 d**
6	📝	Architecture	15 d
7	📝	Functional specification	10 d
8		Security Architecture	5 d
9		**Life-cycle support**	**8 d**
10	📝	Life-cycle model	3 d
11		Security measures	5 d
12		Configuration Management	3 d
13		Delivery Procedures	5 d
14		**Implementation**	**30 d**
15		**Development**	**30 d**
16	📝	Implementation	30 d
17		**Tests**	**7 d**
18	📝	Test plans	7 d
19		**Testing and Integration**	**41 d**
20		**Tests**	**40 d**
21		Testing	30 d
22	📝	Test results	5 d
23	📝	Analysis of coverage	5 d
24		**Guidance**	**41 d**
25	📝	Installation guides	10 d
26	📝	User guidances	10 d
27		Product (TOE) ready	1 d
28		**Operation and Maintenance**	**22 d**
29	📝	Independent testing	2 d
30	📝	Vulnerability Analysis	20 d

Project: Common Criteria schedule
Date: 07.05.10

Figure 12. Common Criteria process in a project

4.3 The Deliverables of Common Criteria at EAL3

This section describes the assurance classes more in detail for EAL3 in terms of its deliverables and their requirements. The classes are examined in the order they are produced as described in the previous chapter. This thesis explains the main targets and requirements of each component. A full description of requirements is seen in Common Criteria standard part 3 (EAL3 requirements are collected in this thesis into Appendix 1) and the Guidelines for Developer Documentation (according to Common Criteria Version 3.1). Common Criteria defines quite strictly the form of documentation of the Protection Profile and Security Target, but not for the other components. This is the reason why the Security Target is examined more carefully compared to other component requirements.

4.3.1 Protection Profile and Introduction Document

The Protection Profile (PP) is an optional document that describes the type of TOE, for instance, firewalls. The Protection Profile can be created by the customer to define the wanted security needs from a product. There are also ready-made Protection Profiles of a TOE type which can be used as a template to create a compulsory Security Target document. In March 2010, there are over 160 evaluated Protection Profile documents in the Common Criteria Portal. If a Security Target uses a PP directly, it is said to be conformant to the named PP.

It is also recommended that an Introduction document is created. The introduction document lists all the requirements and their components (as a standard notation) of the evaluated product with the information (e.g. link) about the document and the section in which the wanted requirement can be found. This way the evaluation and possible updates to the documents are easier to make, while the information where to find the information is in a single document. Even though the evaluation is done only for the exact version of the TOE, the introduction document helps also in planning and developing a new Target of Evaluation by giving a good guideline how the CC requirements can be fulfilled in the organization.

4.3.2 Security Target (ASE –components)

The Security Target is the main document of the Common Criteria requirements. It is a product-specific document describing the evaluated product (TOE) and its security functionalities. It is created first in the project and used as a “requirement specification” for the product to be developed and the process to be applied. The Common Criteria standard defines quite strictly the structure of the ST. Figure 13 shows the content and order of the chapters defines by the CC standard.

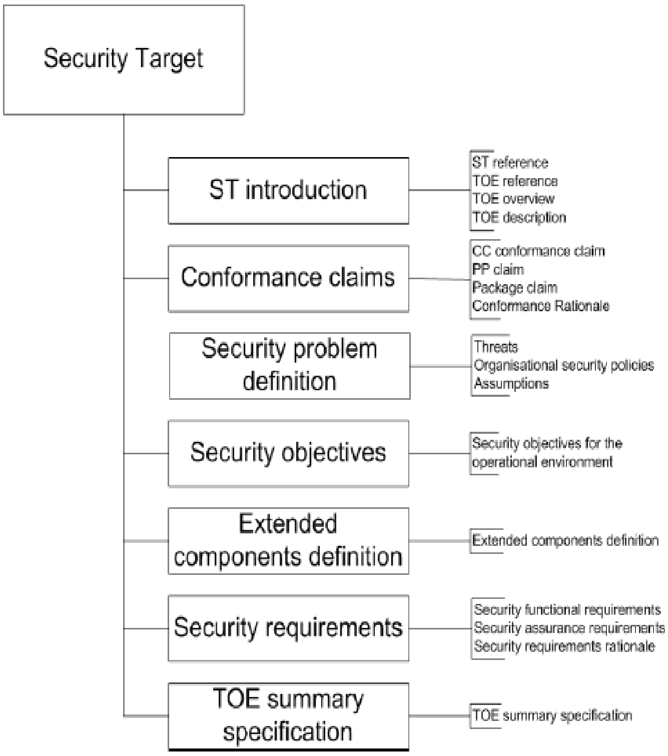

Security Target
ST introduction
ST reference
TOE reference
TOE overview
TOE description
Conformance claims
CC conformance claim
PP claim
Package claim
Conformance Rationale
Security problem definition
Threats
Organisational security policies
Assumptions
Security objectives
Security objectives for the operational environment
Extended components definition
Extended components definition
Security requirements
Security functional requirements
Security assurance requirements
Security requirements rationale
TOE summary specification
TOE summary specification

Figure 13. Security Target content (CC part 1, 2009)

The first stage of the Security Target, ST Introduction, explains the evaluated product and its main security features to a customer. The Target of Evaluation (TOE), which can be only a part of the whole developed product, is defined in the first chapter. It sets the development and documentation requirements for the whole development process of the product. The second chapter, Conformance Claims, lists quite briefly the version and revision of Common Criteria standard used, which Evaluation Assurance Level will be used and if a Protection Profile is conformant with the ST. The third chapter, Security problem definition, defines possible security threats for the TOE, operational security policies (such as laws and the organization’s security policies) and assumptions for the operational environment in which the TOE operates. Chapter four, Security Objectives, lists requirements for the TOE and its operational environment that the threats from chapter 3 can be avoided. The fifth chapter Extended Components Definition explains if there are any security requirements for the Security Target outside the Common Criteria part two and three, for example, any special customer requirements. Chapter 6 Security Requirements selects the suitable and wanted functional requirement components

(SFRs) from the CC part 2 and the assurance requirement components from the CC part 3 for the TOE (see section 3.3). The last chapter TOE Summary specification provides the general technical information of the TOE and summarizes how the SFRs are satisfied by the TOE.

Security Target document moves on systematically from the beginning to the end with a target to show that all the threats for the TOE will be encountered by the set objectives. However, the content of the document is quite heavy (usually a ST is between 50-100 pages) with all the tracing and defining with the notation of CC. Figure 14 is constructed to help to understand the system, and easier to follow-up the chapters of a ST for the reader. The circles in the picture tell the number of the chapter in the Security Target, the boxes the content of the chapter and the arrows the way the contents are traced. The first chapter describes the Target of Evaluation from which the assets to protect can be defined. Using as a base the TOE and assets, chapter 3 lists threats, organizational security policies and assumptions. Chapter 4 defines objectives for TOE and its operational environment. Chapter 4.3 traces that all the listed objectives in chapter 4 encounter one or more listed threats, assumptions or organizational security policies (OSPs) from the chapter 3 and explains the rationale behind it, explaining the justification for the tracing. Chapter 6 lists the security functional requirements (SFRs) and shows tracing in the chapter 6.2 that every SFR addresses one or more objects listed in the chapter 4. Chapter 6.2 also describes the justification that the shown objectives for the TOE are effectively addressed by the SFRs. Finally, in the chapter 7 the TOE security functions (TSFs) are summarized and the tracing back to the SFRs is shown.

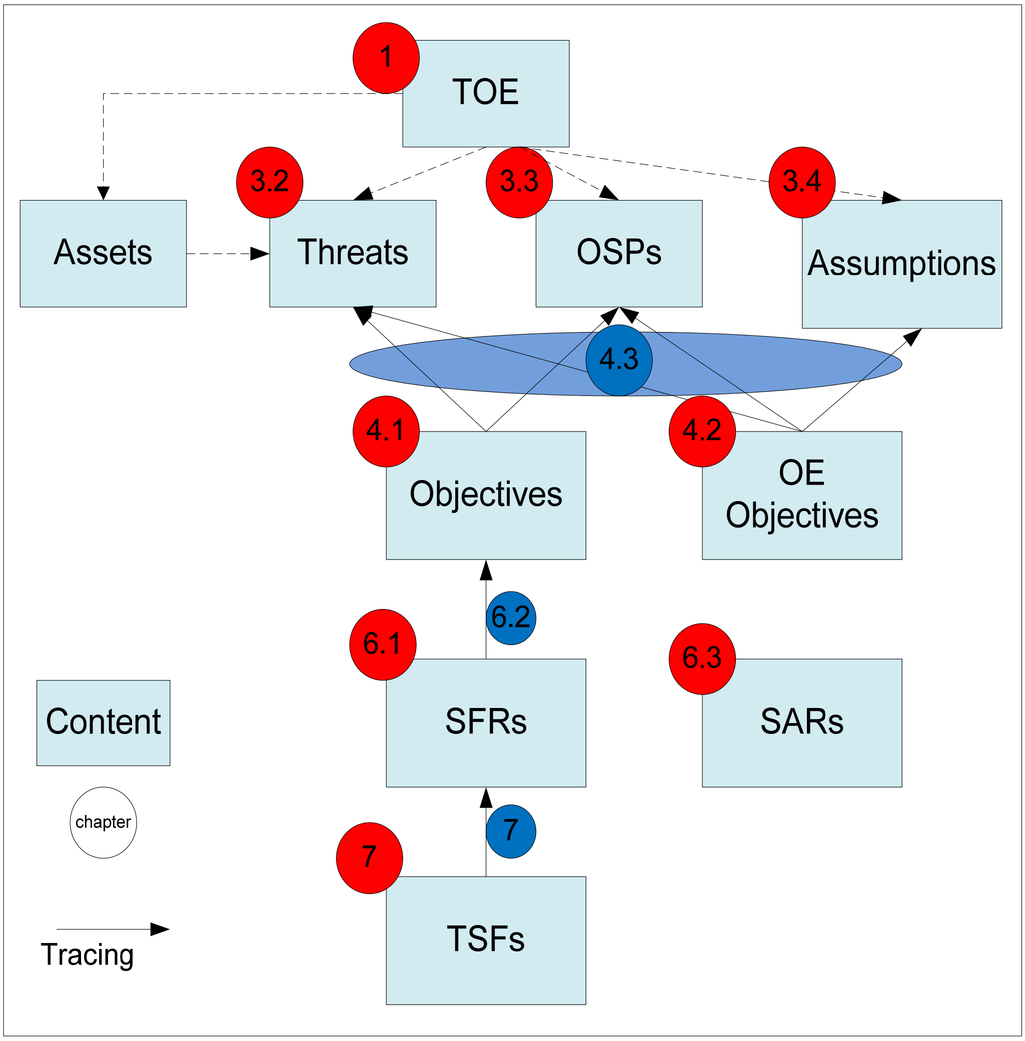


Figure 14. Sequence of Security Target content

Points to consider in ST

Security Target is the document that will be stored also in the Common Criteria portal with the certificate, so it'll be the document the customers and competitors read. The ST sets the requirements for the product and for the process by defining the TOE, TSF and evaluation assurance level to be reached. That's why the careful planning of each step while writing the ST is required. A proper understanding of the CC standard and the technical security features of the product must be high by the writer of the ST. As an example, there are about 200 security functional components in CC part 2 from which usually 10-20 suitable components are chosen. Also every listed point of threats, assumptions, among other things, must be considered, while affecting the next chapters of ST as well as the development and documentation of other requirements.

4.3.3 Life-cycle Support (ALC-components)

Configuration Management (ALC_CMC.3, ALC_CMS.3)

Common Criteria configuration management (CM) sets requirements for CM capabilities and the scope of a CM system. Simplified, a labeled TOE, a configuration management plan and a configuration list (baseline) have to be provided, and a defined CM system must be used.

The label and references of the TOE must be unique and match with the information in the Security Target. A configuration system includes procedures and tools used by the developers, and the system must distinguish different versions and the status (e.g. draft, final) of the configuration items. In EAL3 the CM system does not have to be automated.

An overview of the CM system and a description of how to use it must be documented. The CM documentation must also describe the access control measures so that only authorized changes are made to the configuration items. A CM plan of the project (part of CM documentation) describes how the CM system is used for the development of the TOE.

Configuration management scope requires a configuration list as a deliverable. The CM list must include the TOE, the software modules and hardware components, TOE implementation presentation with all the deliverable documents in EAL3. All items in the list shall have a unique identification. In addition, for each TOE Security Function (TSF) relevant configuration item, the list shall include the name of the developer.

Defined Life-cycle Model (ALC_LCD.1)

The purpose of the life-cycle definition family is to use a defined life-cycle model in developing and maintaining the TOE, in order to minimize the likelihood of security flaws. Common Criteria, nevertheless, does not require using any standard life-cycle models. The documented development model must be used during the development by the developers and the model shall have the necessary controls over the development and maintenance of the TOE.

The documentation should include information about life-cycle phases and boundaries between the subsequent phases. For each life-cycle phase the document shall describe the activities within the phase, the relationship to other phases, procedures, tools and techniques, roles and responsibilities, and possible involvement of third parties.

Security Measures (ALC_DVS.1)

For the Security Measures family the developers have to provide documentation describing all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment. The documentation has to specify the policies for confidentiality and integrity.

The security aspects to be covered in the documentation are organization, development organization, technical development environment, security policies, personnel security, access control, transfer of protected material and security management.

An audit is held by the evaluator to check that the procedures are followed as documented.

Delivery Procedures (ALC_DEL.1)

Developers have to document the delivery procedure of the finished TOE to the customer. The CC standard defines no mandatory specific delivery practices. The target of the delivery procedure is to ensure the secure transfer to the customer and determine the identification of the TOE. Secure delivery can be achieved, for example, by a sealed envelope or tamper proofed seals, cryptographic checksums or encryption to assure confidentiality. The necessity of the secure procedure is evaluated by the chosen level of the vulnerability assessment (see the section 4.3.7).

4.3.4 Development (ADV–components)

Architectural Design (ADV_TDS.2)

The architectural design describes the design of the TOE in a way that the realization of the security functional requirements can be noted. At EAL3, the design structure of the TOE is described in terms of the subsystems (or modules).

The next requirement is to identify the subsystems which are composed of TOE Security Functionalities. The subsystems are divided into SFR-non-interfering, SFR-enforcing and SFR-supporting subsystems. A SFR-non-interfering subsystem is not dependent upon implementing an SFR. The behaviors of these subsystems are described only in a way that it is visible that it doesn't have any SFRs. A SFR-enforcing subsystem implements mechanisms of a security functional requirement. The behavior of these subsystems is described more in detailed in terms of how a subsystem provides the security functionality as well as a summary of other features. A SFR-supporting subsystem is dependent upon a SFR-enforcing subsystem but does not have a direct role in a SFR-supporting requirement. The behavior of these subsystems must be summarized. Even though the standard categorizes the subsystems in the mentioned way, the developer does not have to categorize the subsystems in his documentation, only to provide necessary information of the subsystems.

The design documentation shall also describe the interactions among all subsystems of the TSF. The goal of this is to help the reader to understand better how the TSF performs its functions. The last requirement of the architectural design documentation is to provide a mapping showing that all TSF interfaces (from Functional Specification) trace to the described subsystems.

Functional Specification with Complete Summary (ADV_FSP.3)

The Functional specification class sets requirements for the TOE Security Functionality Interface (TSFI) descriptions. The TSFI consists of invoking a service from the TSF and corresponding responses to the invocations. The TSF processes are described in the Architectural Design documents and the description is not needed here.

Similar to architectural design, there are SFR-enforcing, SFR-supporting and SFR non-interfering interfaces. An interface is SFR-enforcing, if an action through an interface can be traced to an SFR (from ST). An interface is SFR-supporting, if an SFR-enforcing functionality depends on the interface, but needs to function correctly in order for the security policies of the TOE to be preserved. If an interface has no dependence on SFR-enforcing functionality, it is SFR-non-interfering.

The main requirement for the developer is to provide a functional specification describing the functionality and method of use of all TSF Interfaces. The TSFs are described in the Security Target. For each TSFI, all parameters must be described accurately.

For SFR-enforcing TSFIs, the SFR-enforcing actions by the TSFI must be described as well as direct error messages resulting from invocation of the interface. The non-SFR-enforcing actions of each TSFI are summarized in a way that the evaluator can analyze the TSF Interfaces. However, the developer does not have to label the interfaces according to these categories.

Lastly, the developer must provide a tracing showing that the SFRs (from the Security Target) trace to the TSFIs in the functional specification.

Security Architecture Description (ADV_ARC.1)

The Security Architecture family sets requirements for the security principles and how these are supported in the TOE. The first requirement is to design and implement the TOE in a way that the security features cannot be bypassed. The security principles self-protection, domain isolation and non-bypassability must be designed, implemented and documented.

The Security Architecture description shall be at the same level with Architectural Design documentation (ADV_TDS.2) and describes the security domains (environments that supplied by the TSF) maintained by the TSF. The Security Architecture description shall demonstrate how the TSF initialization process preserves security by listing the system initialization components and their processes from down state to a secure stage. The documentation shall also demonstrate the self-protection, how the TSF protects itself from tampering, and demonstrate the preventing of bypass of the SFR-enforcing functionality.

The evaluator checks that all TSFIs are analyzed and the demonstration shows that no security flaws are found from the system.

4.3.5 Tests (ATE-components)

Functional Testing (ATE_FUN.1)

Functional testing component assures that the tests are performed and documented correctly. The objective of functional testing is to test the TSFs (TOE Security Functionalities).

Test documentation shall include test plans, expected test results and actual test results. The test plan identifies the TOE to be tested and the tests to be performed. The test descriptions shall include possible test pre-requisites and test procedure description with the information about test execution, inputs, expected results with anticipated outputs of successful execution and cleanup process. The test documentation has to include descriptions of the behavior of TSF subsystems and their interactions (and to be consistent with design documentation). The test documentation shall also describe the actual test results and demonstrate that they are consistent with the expected test results.

Testing: Basic Design and Analysis of Coverage (ATE_DPT.1, ATE_COV.2)

The depth of testing and the Analysis of coverage are explained here in the same section because of their similarities and their dependencies to development and functional testing documentation.

The meaning of the Basic testing component is to provide the analysis of the depth of testing. The analysis shall demonstrate that all TSF subsystems in the TOE have been tested.

The purpose of the Analysis of coverage component is to assure that the TSF has been tested against its functional specification. The analysis shall show the correspondence between the tests in the functional test documentation and the TSFIs in the functional specification and to conclude that all the TSFIs have been tested.

For both the cases the correspondence can be proven by a matrix and the analysis of coverage can be made by referring to the matrix and briefly analyzing the coverage in text. Table 7 shows an example of the template of the matrix that can be used.

Table 7. Analysis of test coverage

TSFI / Subsystem	Tests	Analysis of coverage
TSFI-1, secure functionality 1	T1, T2, …	
TSFI-1, secure functionality 2	…	
…	…	
TSFI-2, secure functionality 1	…	
	…	
Subsystem 1, behavior 1	T3	
Subsystem 1, behavior 2	T4	
…		

Independent Testing –sample (ATE_IND.2)

The independent testing is done by the evaluator. The target of this component is to show that the TOE functions as stated by the CC documentation. For that the developer must provide the TOE for testing (the TOE shall be suitable for testing). In addition the developer must provide an equivalent set of resources that were used in the developer's functional testing of the TSF that the evaluator can repeat some tests made by the developers.

4.3.6 Guidance Documents (AGD-components)

Preparative Procedures (AGD_PRE.1)

The meaning of Preparative procedures is to provide guidance about the secure acceptance and installation of the TOE. The acceptance procedures shall describe all the steps in secure acceptance. This guidance must be in line with the delivery procedures described. As a minimum, the procedures must contain that a user checks all the parts and correct versions of the TOE received. In addition, there should be a description if a user can verify the integrity and authority of the TOE and if there's a possibility that a user could detect non-authorized delivery and how to proceed then.

The installation guidance shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment. The objectives for operational environment described in the Security Target have to be in line with installation guidance.

Operational User Guidance (AGD_OPE.1)

Operational user guidance gives instructions to users how to use the TOE in a secure manner. The user guidance can be divided into a general and user role specific description. The general description shall identify the different user roles and describe all possible modes of operation of the TOE.

The user role specific description can be divided into four main requirements. The description of secure functions shall identify for each user role the visible security interfaces and the methods by which the interface is invoked, as well as the descriptions of privileges of each user role and warnings regarding the use of functions and privileges. The description of interfaces shall describe the user-accessible interface parameters to be set by the user, their purpose (in terms of secure and insecure usage) and the immediate TOE response. The event section shall present each type of security-relevant event (e.g. system crash, audit trail overflow) and what are the possible actions a user has to make in order to maintain security. The fourth requirement for user specific guidance is to describe the security measures in order to fulfill the security objectives for the operational environment. The last section has to be checked to be in line with the Security Target. The evaluator also checks that the operational user guidance material is consistent with the functional specification and other development documents.

4.3.7 Vulnerability Analysis (AVA_VAN)

Vulnerability analysis determines the possibility of exploitable vulnerabilities in the development or in the operation of the TOE. Vulnerability assessment is done by the evaluator. The developer has to provide the TOE and the CC documentation of the TOE for the analysis.

The evaluator confirms that the documentation meets the requirements, performs vulnerability analysis by a search of public domain sources to identify potential vulnerabilities in the TOE and by analysis of the TOE using the CC documentation of the TOE. The evaluator also performs penetration testing, based on the identified potential vulnerabilities, to confirm that the TOE is resistant to attacks. The level of the TOE to be resistant to attackers at EAL3 is called Basic. The exact criteria for how to calculate the value of potential attack is provided for the evaluators in Evaluation methodology

(CC CEM, 2009, page 421). The factors influencing the result of the attack potential required to exploit the vulnerability are elapsed time, specialist expertise, knowledge of the TOE, window of opportunity and IT hardware/software or other equipment required for exploitation.

4.3.8 Maintenance, Evaluation and Certification

Usually during the development and evaluation process, several versions of documents have to be created in order to achieve the requirements. This is why the proper configuration management and maintenance process of documents is important. The evaluator provides an intermediate report of each activity by stating a verdict *success, fail* or *inconclusive* (when more information is needed to make a verdict). Intermediate reports are collected into a final report by the evaluator, and the certification bodies can make the decision whether or not the verdict is success. The evaluation and certification is done to the exact version of the product and in the case of an update to a newer version of a product, a completely new evaluation must be performed. However, when the introduction document is done well, it may be that some of the deliverables need only to be checked quickly to be still valid.

5. Results

The result section collects and summarizes the lessons learnt from the case study and from the relevant scientific literature. The first part discusses the Common Criteria from the developer’s point of view, explaining the findings arising from the case study. The second part discusses the Common Criteria from the evaluation’s point of view, analyzing the opinions mainly arising from the literature. The third part adds the customer’s point of view, listing the benefits of the CC to the customers. The fourth part discusses the current trend of Common Criteria and its possible future prospects for the standard. Finally, all the findings are summarized in terms of SWOT analysis, listing Common Criteria's current strengths (S) and weaknesses (W), and future opportunities (O) and threats (T).

5.1 CC and Developers: Lessons Learnt from the Case Study

Studying of the Common Criteria Standard

The IT product development project was decided to do without an official evaluation but fulfilling all EAL3 requirements performed by developers. The study of the Common Criteria started by examining the official standard material and by reading other available material. A one day training session was ordered from an official evaluator company to get a better understanding of the overview of the standard and actual work needed for evaluation assurance level 3. Some of the assumptions made only by reading the standard before the workshop were refuted, as the Common Criteria uses similar terms as a general development process, but with a different interpretation (e.g. the content in functional specification and functional testing). In this respect, proper training by an official evaluator organization is highly recommended for any organization performing assessment according to the Common Criteria standard for the first time.

Common Criteria in a Development Process

The process of Common Criteria requirements must be done in a consecutive order as the first decisions made in the documentation have effects on the next documents and so on. In this sense, the CC process was easier to adapt to the waterfall model used by the project than, for example, to projects using agile development methods. Moreover, it is also quite important to have the understanding of the CC requirements already right at the beginning of the project. This way the decisions made on security aspects of the product are handled in the right way from the beginning and no changes are needed during the process because of any lack of understanding of the correct security requirements of the CC. A suggestion how the CC requirements with its documentation are added to a project's process appropriate phase (when each CC deliverable should be done in the development process) is shown in Appendix 2 and explained in more detail in Section 4.2.

Knowledge Needed by the Developers

The knowledge pool of the developers has to be wide. In the beginning, the writer of the Security Target needs to know well the architectural and technical aspects of the product, the security aspects of IT products and the Common Criteria standard. Even though an architecture engineer of the project was very familiar with the product and the security matters, he stated that the structure of a readymade Security Target was very difficult to understand and the tracing was hard to follow.

The writers of the life-cycle documents have to be very familiar with the different processes of the organization. The developers of the product (software and hardware developers, test people and guidance document writers) have to have both the technical knowledge and the Common Criteria understanding.

Since the CC standard is written as "standard language", the requirements were difficult to interpret for the practical work. That's why it would be good to have a Common Criteria expert (who also understands the basics of each process area) explaining to the developers what is the most suitable assurance level to be pursued, what are the components of each level, and what each requirement means in practice. Every developer should also have an understanding of the consequences of his choices to other developers. For example the tester has to properly test the functionalities in a way the software developer has divided the subsystems and described the interfaces. Also all the assumptions described in the Security Target, trace to the guidance documents meant for customers.

Scope of the Evaluated Product

Defining the scope and the Target of Evaluation was quite a difficult task also in this project. In many cases, the Common Criteria requirement and its scope comes from the customers, but in this product development project the future customers who demand the CC evaluation were unknown. Therefore the scope was supposed to be reasonable in terms of the main security features but at the same time not to add too much extra work for the project. The TOE was defined (by making a rough guess) so that it covered more than an average product evaluated by CC. Yet, because of the current well-defined process model of the organization, the extra documentation caused by the Common Criteria was mostly added to the obligatory documents to which, nevertheless, had to be done. A couple of requirement families were already done during the current process model and those documents needed only to be checked so that all the requirements certainly exist. Only the Security Target document, the content and form of which is strictly defined by the CC standard, had to be developed from scratch.

The content of the Security Target document defines, for instance, possible threats for the TOE, organizational policies, and assumptions for the operational environment. In principle it is possible to define, for example, assumptions in such a way that the TOE is assumed to be in an environment where an attacker cannot physically attack. This was considered quite odd, and that’s why it is important also for the readers of the ST to check what kind of assumptions and other requirements for the TOE are chosen in ST.

Subcontractors in a Project –What has to be required from them

When a part of the developed product is subcontracted from another organization, also the work products from the subcontractor have to be taken into account in the Common Criteria requirements. The safest way is to require all the CC requirements to be fulfilled in the contract. However, software subcontractor cases the most sensible way would be to do the Security Target personally or together with the subcontractor and then require precise CC development and test class requirements, tolerant life-cycle deliverables and all required inputs to the guidance documentation.

The Security Target defines the overall security requirements for the product, and therefore it would be good to state personally (also easier for the subcontractor to understand what the customer is expecting from them). Because the CC development requirements cover exactly the implementation of the TOE, it can almost only be done by the actual implementer. The CC test requirements are security specific requirements based on the development documentation and, therefore, they would also be a part of the subcontractor's responsibility. Usually the life-cycle descriptions are very different depending on the organization, and thus the actual CC deliverables have to be performed by the organization itself. However it is still good to check that the subcontractor is also fulfilling all the life-cycle requirements, and notations of the procedures can be added to the actual CC deliverables. The guidance documents are usually made by the organization itself, but inputs to these are needed also from the subcontractors. It is also very important to notice that when the evaluation level is low (EAL1-EAL3), the development documentations required by the Common Criteria are quite general. Therefore, additional deliverables for other purposes could be required from subcontractors.

Best Advantages to the Development Process

One of the most visible things in applying the CC standard was that the security aspects were involved deeper and more systematically during the project's development life-cycle. Some discussions about the implementation of security features may have gone without thinking and brainstorming if the CC standard would have not been in the scope of the project. It was also noticed that the best way to actually accomplish security requirement deliverables within the project is to add the requirements to the current process descriptions.

5.2 CC and Evaluation: Findings from Literature

Pros

Although the official evaluation was not done during this project, the training and other literature material give consistent feedback about the evaluation. The advantage that the evaluation is done in many phases during the project, gives the developers the possibility to improve the product and documentation during the planning and implementation. Also the evaluators can help the developers by guiding them through the process and clarifying the confusions and misunderstandings from reading the CC standard. By performing an evaluation, the developers can also make sure that the IT product fulfills the main security requirements. This evidence of security in a form of a certificate, can give the feeling of certainty for the developer when tendering the IT product to the customer.

Cons

The disadvantage of the evaluation is that the evaluation process is expensive; the CC is approximately 10-40 % of the development costs and it delays the product's time to market (Gollmann, 2006). Therefore usually the scope of the CC evaluation is limited to only a part of the product to minimize the extra work, extra costs and time delay. This sets a contradictory practice to the standard: the evaluation is done to show to the customers that the product is developed to be secure, but at the same time it may only examine a part of the product. Also the chosen EAL expresses only an assurance level achieved by the process, not the exact level of security in the product. Because the evaluation is valid only for one version of the product, the cost of reevaluating new versions of an evaluated product is high (almost as high as for doing a completely new product).

5.3 CC and Customers: What Do Customers Benefit

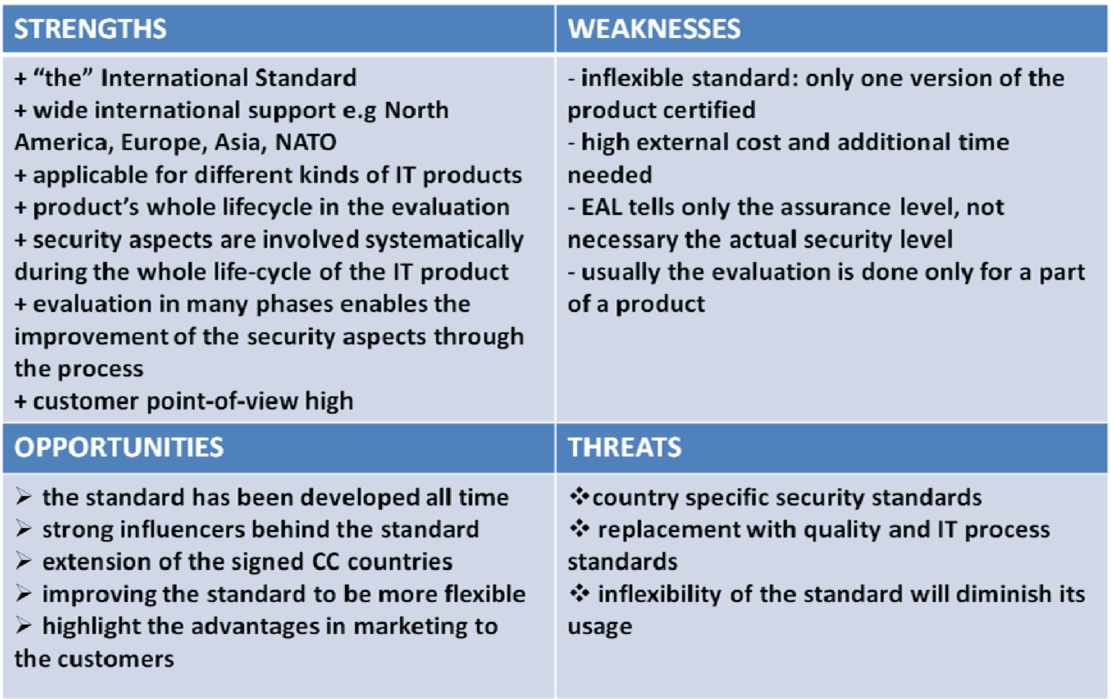
The customer point of view is well taken into account in the CC standard. The Protection Profiles can be written by the customers to define the security requirements of the product they need. Also there are chapters in the Security Target document that are directly written for the customers. Evaluated Security Target documents (may be adapted versions) of a product are saved in the CC portal and they are available for everyone to read. That's also a good way for the customers to look for products they want to purchase or get examples of security requirements they would need for a product. At the moment, the Common Criteria standard is mostly required in some countries by public customers, especially in certain markets such as with smart cards (Gollmann, 2006).

5.4 Trend of CC and Future Prospects

The Common Criteria is still quite a new standard, and it is being developed all the time. The newest version 3.1 revision 3 was published in July 2009 and the next version is already under development. There are a lot of influential signed countries involved in the Common Criteria, and because of international agreements, there are lot of possibilities and good signs to the future. As the Common Criteria is a generic IT security standard, it can be used for all possible IT products. However, most of the research material about the Common Criteria found in the literature research for this thesis is done before 2005, so the current interest in CC seems to be quite low. The time, expense and inflexibility of Common Criteria evaluation are causing its appeal to flag, and there are dynamic accreditations that find favor over CC evaluation. This issue was addressed at the NATO IA symposium (held in 22-24.9.2009) by a senior NATO official supporting the need for a more flexible and timely approach (Nexor, 2009). Also national security standards, other security process models and quality standards are sometimes considered as an alternative to the Common Criteria. The advantage compared to the national standards is that a CC certified product is applicable in all the other countries too. The general security processes describe usually the life-cycle models but not the product itself whereas the Common Criteria tackles both aspects. Although quality standards increase the overall quality of the procedures, the actual security mindset is missing from them.

5.5 SWOT – Analysis

The results discussed in this chapter are summarized in Figure 15. The aspects are analyzed by SWOT-analysis, categorizing the findings in terms of current strengths and weaknesses, and future opportunities and threats.



STRENGTHS	WEAKNESSES
+ “the” International Standard + wide international support e.g North America, Europe, Asia, NATO + applicable for different kinds of IT products + product’s whole lifecycle in the evaluation + security aspects are involved systematically during the whole life-cycle of the IT product + evaluation in many phases enables the improvement of the security aspects through the process + customer point-of-view high	- inflexible standard: only one version of the product certified - high external cost and additional time needed - EAL tells only the assurance level, not necessary the actual security level - usually the evaluation is done only for a part of a product
OPPORTUNITIES	**THREATS**
➢ the standard has been developed all time ➢ strong influencers behind the standard ➢ extension of the signed CC countries ➢ improving the standard to be more flexible ➢ highlight the advantages in marketing to the customers	❖country specific security standards ❖ replacement with quality and IT process standards ❖ inflexibility of the standard will diminish its usage

Figure 15. SWOT analysis of Common Criteria

When analyzing the Common Criteria and its usage, there are more strengths than weaknesses, as well as more opportunities than threats. The strengths summarize the Common Criteria's wide use, both in terms of international support and in terms of standard's content. The weaknesses criticize the Common Criteria's inflexibility and lack of veracity. Opportunities highlight the developing of the standard to a more attractive model for different stakeholders. Threats notice the possibility of replacing the Common Criteria with other standards.

6. Conclusion

Information technology security has become an important factor in our everyday life in ensuring the correctness of the IT systems we use. There are various methods and tools available to protect the IT security from possible attacks, and new advanced methods are being developed all the time. IT security standards are one way to support the security tools, to assure secure development of the products, and to certify a proper level of security for products and processes.

The starting point for this thesis was to examine the international IT security standard called the Common Criteria. The main research questions set in the beginning of the study were to find out how the Common Criteria standard requirements can be applied in a product development process and what are the benefits and weaknesses of the standard.

The research was made based on the literature research and on a case study using Common Criteria's evaluation assurance level 3 requirements. It was noticed that

1) The Common Criteria sets requirements for the whole life-cycle process of the product, and these requirements can be added quite smoothly to IT development projects using waterfall model.
2) Although implementing the CC requirements for the first time added extra workload to the project, there were more discussions about the security related matters of the product and visible proposals for improvements that could have been forgotten without a "compulsory requirement".
3) Proceeding according to the CC standard gives confidence about the security aspects in the product both for developers and customers, but the schedule and work amounts of the evaluation have to be considered already in the customer’s bid phase.

The Common Criteria standard has a wide international support and it has been developed continuously from its origins at the end of 1990s (when three standards together were combined into a single standard) to today. However, its inflexibility mainly in terms of time, expense and certification scope has brought up questions if a more dynamic standard is needed with an easier maintenance process of the certificate to replace the Common Criteria.

Today, as well as in the future, customers will require IT security standards to ensure the secureness of the IT system or product. In addition, the standards are needed to support the developers to have defined procedures for the secure development of products. Further studies are needed to assess the next developed versions of the Common Criteria standard and how the standard will answer to the expected changes in order to grow for a more dynamic standard.

References

Anderson Ross (2001). Security Engineer: A Guide to Building Dependable Distributed Systems. USA: John Wiley & Sons, Inc.

Asymmetric cryptography: Asymmetric Cryptography Information & Resources, last modified 04.04.2007. Available: http://asymmetriccryptography.com/ (accessed 28.1.2010)

Austin R., Lyytinen K., Penttinen E., Saarinen T., Applegate L (2009). F-Secure Corporation: Software as a Service (SaaS) in the Security Solutions Market. Harvard Business School, rev: February 26, 2009, 9-809-099

BEA Systems. B2B security: Implementing Nonrepudiation. Available: http://download.oracle.com/docs/cd/E13214_01/wli/docs70/b2bsecur/nonrep.htm#1036137 (accessed 5.2.2010)

Beissinger, Janet; Pless, Vera (2006). Cryptoclub : Using Mathematics to Make and Break Secret Codes. Natick, MA, USA: A K Peters, Limited, 2006. p 4. Available: http://site.ebrary.com/lib/aalto/Doc?id=10160959&ppg=21 (accessed 5.2.2010)

CC CEM (2009). Common Criteria (2009): Evaluation methodology. Version 3.1, revision 3. Available: http://www.commoncriteriaportal.org/thecc.html (accessed 28.06.2010)

CC part 1 (2009). Common Criteria (2009): Part 1 Introduction and general model. Version 3.1, revision 3. Available: http://www.commoncriteriaportal.org/thecc.html (accessed 12.02.2010-15.5.2010)

CC part 2 (2009). Common Criteria (2009): Part 2 Security functional requirements. Version 3.1, rev 3. Available: http://www.commoncriteriaportal.org/thecc.html (accessed 12.02.2010-15.5.2010)

CC part 3 (2009). Common Criteria (2009): Part 3 Security assurance requirements. Version 3.1, revision 3. Available: http://www.commoncriteriaportal.org/thecc.html (accessed 12.02.2010-15.5.2010)

CC Developers Guide (2009): Guidelines for Developer Documentation. Documentation according to Common Criteria Version 3.1 (2009). Available: http://www.commoncriteriaportal.org/files/ccfiles/CommonCriteriaDevelopersGuide_1_0.pdf (accessed 19.2.2010)

Chaffey Dave, Wood Steve (2005). Business Information Management. Improving Performance Using Information Systems. Pages 355-356. Prentice Hall.

Common Criteria Portal. Available: www.commoncriteriaportal.org (accessed 5.2.2010 -15.5.2010)

Conkling, W. R. and Hamilton, J. A. (2008). The importance of information security spending: an economic approach. InProceedings of the 2008 Spring Simulation Multiconference (Ottawa, Canada, April 14 - 17, 2008). Spring Simulation Multiconference. Society for Computer Simulation International, San Diego, CA, 293-300. Available:
http://portal.acm.org.libproxy.tkk.fi/citation.cfm?id=1400549.1400590&coll=portal&dl=ACM&CFID=1784450&CFTOKEN=66607831# (accessed 27.1.2010)

CSI Computer Crime and Security Survey (2009). Main findings. Available http://www.gocsi.com/2009survey/ (accessed 27.1.2010)

Dent, Alex; Mitchell, Chris (2004). User's Guide to Crytography and Standards . Norwood, MA, USA: Artech House, Incorporated, 2004. p 159-165. Available: http://site.ebrary.com/lib/aalto/Doc?id=10082005&ppg=175 (accessed: 5.2.2010)

Digitoday (2010), Saksa varoitti Internet Explorerista.
http://www.digitoday.fi/tietoturva/2010/01/18/saksa-varoitti-internet-explorerista/2010696/66 (accessed 22.1.2010)

Gollmann Dieter (2006). Computer Security second edition. West-Sussex, England: John Wiley & Sons, Ltd.

Herrmann, Debra S. (2003). Using the Common Criteria for IT Security Evaluation. Auerbach.

ISO (2010). International Organization for Standardization. ISO Standards for Information Technology ISO/IEC 19790:2006 and ISO/IEC 24759:2008. Available:
http://www.iso.org/iso/catalogue_detail.htm?csnumber=33928,
http://www.iso.org/iso/catalogue_detail.htm?csnumber=41529 (accessed 13.4.2010)

Kennedy T, John Manuel. Information Security Components Figure. Available at: http://en.wikipedia.org/wiki/Information_security (accessed 29.4.2010)

Kioskea, Electronic signatures. Last modified October 16, 2008. Available: http://en.kioskea.net/contents/crypto/signature.php3. (accessed 04.02.2010)

Koskinen J, 2001. TTKK, Tietoturvallisuuden perusteet. Tietoturvaprotokollia. Available: http://www.cs.tut.fi/kurssit/8306000/2001/pr.html (accessed 5.4.2010)

Kurose James F., Ross Keith W. (2005). Computer networking: A top-down approach featuring the Internet. 3rd edition. Pearson education. Pages 654-701

Lendering Jona (2010). Articles of Ancient history: Gaius Julius Caesar. Available: http://www.livius.org/caa-can/caesar/caesar00.html (accessed 5.2.2010)

McCormack John, Conway Damian (2005). Software development process. CSE2305 – Object-Oriented Software Engineering course material, Monash University, School of Computer Science and Software Engineering. Available: http://www.csse.monash.edu.au/~jonmc/CSE2305/Topics/07.13.SWEng1/html/text.html (accessed 3.3.2010)

Mellado Daniel, Fernandez-Medina Eduardo, Piattini Mario (2007). A common criteria based security requirements engineering process for the development of secure information systems. Computer Standards & Interfaces, Volume 29, Issue 2, February 2007, Pages 244-253. ISSN 0920-5489, DOI: 10.1016/j.csi.2006.04.002. Available: http://www.sciencedirect.com/science/article/B6TYV-4K4WH4K-1/2/95b9f9bc6a5b06873e836da933e127a3 (accessed 11.2.2010)

Nexor (2009). Dynamic Accrediation Finds Favour Over Common Criteria Evaluation. Available: http://www.nexor.com/headlines/accreditation (accessed 3.4.2010)

Nieminen, M. (2010). Customer’s of Nordea as a victim of malware, police is investigating the subject. Helsingin Sanomat electronic version. 16.1.2010 Available: http://www.hs.fi/talous/artikkeli/Nordean+asiakkaita+haittaohjelman+uhriksi+poliisi+tutkii+asiaa/1135252180911 (accessed 22.1.2010).

NIST (2010). National Institute of Standards and Technology: FIPS PUB 140-2 Standard. Available: http://csrc.nist.gov/groups/STM/cmvp/standards.html#02 (accessed 31.3.2010)

Nordea (2010). Loss caused by malware will be compensated. Nordea webpages [Online]. Available: http://www.nordea.fi/About+Nordea/Loss+caused+by+malware+will+be+compensated/1286002.html (accessed 22.1.2010).

Oppida (2010). Common Criteria Evaluation & Certification and Development of Evidence for a CC v 3.1 Evaluation. Version 4. Training material provided by Oppida on 10.2.2010.

Rehman, S. and Mustafa, K. (2009). Research on software design level security vulnerabilities. SIGSOFT Softw. Eng. Notes 34, 6 (Dec. 2009), 1-5. Available: http://doi.acm.org/10.1145/1640162.1640171 (accessed 22.1.2010)

Richardson, R. (2007). CSI Computer Crime and Security Survey. CSI Survey 2007. Available: http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pdf (accessed 27.1.2010)

Rinne (2010). Simo Rinne, IT security specialist. Conversations about IT security in PSS networks on 29.3.2010 and 14.5.2010 in Helsinki.

RSA Security (2010). Information Security Glossary. Available: http://www.rsa.com/glossary/default.asp?id=1080 (accessed 5.4.2010)

SSE-CMM (2006). The Systems Security Engineering Capability Maturity Model. Available: http://www.sse-cmm.org/index.html (accessed 1.4.2010)

TKK Tietoturvallisuustekniikka (2008). Helsinki University of Technology, T-110.4200 Tietoturvallisuustekniikka. Lecture Slides 12, 2008. Available: www.tml.tkk.fi/Opinnot/T-110.4200/2008/Lectures/12-Audit-People.pdf (accessed 24.3.2010)

U.S code collection: Definitions. Cornell University Law School http://www.law.cornell.edu/uscode/44/3542.html (accessed 22.1.2010)

Vargas Enrique, 2000. High Availability Fundamentals. Sun BluePrints^{TM} Online – November 2000. Available: http://www.sun.com/blueprints/1100/HAFund.pdf (accessed 08.04.2010)

Appendices

Appendix 1. Common Criteria EAL3 Requirements

The Common Criteria requirement families and components at the evaluation assurance level 3 are collected from the Common Criteria part 3 to this appendix.

ADV_ARC.1 Security architecture description:

Dependencies: ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFR

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_FSP.3 Functional specification with complete summary

Dependencies: ADV_TDS.1 Basic design

Developer action elements:

ADV_FSP.3.1D The developer shall provide a functional specification.

ADV_FSP.3.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.3.1C The functional specification shall completely represent the TSF.

ADV_FSP.3.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.3.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.3.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.3.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from SFR-enforcing actions and exceptions associated with invocation of the TSFI.

ADV_FSP.3.6C The functional specification shall summarise the SFR-supporting and SFR-non-interfering actions associated with each TSFI.

ADV_FSP.3.7C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_TDS.2 Architectural design

Dependencies: ADV_FSP.3 Functional specification with complete summary

Developer action elements:

ADV_TDS.2.1D The developer shall provide the design of the TOE.

ADV_TDS.2.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation elements:

ADV_TDS.2.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.2.2C The design shall identify all subsystems of the TSF.

ADV_TDS.2.3C The design shall describe the behaviour of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.

ADV_TDS.2.4C The design shall **describe** the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.2.5C The design shall summarise the **SFR-supporting and SFR-non-interfering** behaviour of the SFR-enforcing subsystems.

ADV_TDS.2.6C The design shall summarise the behaviour of the **SFR-supporting** subsystems.

ADV_TDS.2.7C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.2.8C The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

ALC_CMC.3 Authorisation controls

Dependencies: ALC_CMS.1 TOE CM coverage
ALC_DVS.1 Identification of security measures
ALC_LCD.1 Developer defined life-cycle model

Objectives

327 A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labelling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

328 Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE.

329 The use of a CM system increases assurance that the configuration items are maintained in a controlled manner.

330 Providing controls to ensure that unauthorised modifications are not made to the TOE ("CM access control"), and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE.

Developer action elements:

ALC_CMC.3.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.3.2D The developer shall provide the CM documentation.

ALC_CMC.3.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.3.1C The TOE shall be labelled with its unique reference.

ALC_CMC.3.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.3.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.3.4C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ALC_CMC.3.5C The CM documentation shall include a CM plan.

ALC_CMC.3.6C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.3.7C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.3.8C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMS.3 Implementation representation CM coverage

Dependencies: No dependencies.

Objectives

355 A CM system can control changes only to those items that have been placed under CM (i.e., the configuration items identified in the configuration list). Placing the TOE itself, the parts that comprise the TOE, the TOE implementation representation and the evaluation evidence required by the other SARs under CM provides assurance that they have been modified in a controlled manner with proper authorisations.

Application notes

356 ALC_CMS.3.1C introduces the requirement that the TOE implementation representation be included in the list of configuration items and hence be subject to the CM requirements of CM capabilities (ALC_CMC).

Developer action elements:

ALC_CMS.3.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.3.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs (*Security Assurance Requirements, part 3*); the parts that comprise the TOE; and the implementation representation.

ALC_CMS.3.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DVS.1 Identification of security measures

Dependencies: No dependencies.

Developer action elements:

ALC_DVS.1.1D The developer shall produce and provide development security documentation.

Content and presentation elements:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

Content and presentation elements:

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ASE: Security Target Evaluation

ASE_CCL.1 Conformance claims

Dependencies: ASE_INT.1 ST introduction
ASE_ECD.1 Extended components definition
ASE_REQ.1 Stated security requirements

Developer action elements:

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

Content and presentation elements:

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

ASE_ECD.1 Extended components definition

Dependencies: No dependencies.

Developer action elements:

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements:

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_INT.1 ST introduction

Dependencies: No dependencies.

<u>Developer action elements:</u>

ASE_INT.1.1D The developer shall provide an ST introduction.

<u>Content and presentation elements:</u>

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_OBJ.2 Security objectives

Dependencies: ASE_SPD.1 Security problem definition

<u>Developer action elements:</u>

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

<u>Content and presentation elements:</u>

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment. Class ASE: Security Target evaluation

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_REQ.2 Derived security requirements

Dependencies: ASE_OBJ.2 Security objectives
ASE_ECD.1 Extended components definition

<u>Developer action elements:</u>

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

<u>Content and presentation elements:</u>

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_SPD.1 Security problem definition

Dependencies: No dependencies.

Developer action elements:

ASE_SPD.1.1D The developer shall provide a security problem definition.

Content and presentation elements:

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_TSS.1 TOE summary specification

Dependencies: ASE_INT.1 ST introduction

ASE_REQ.1 Stated security requirements

ADV_FSP.1 Basic functional specification

Developer action elements:

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

Content and presentation elements:

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ATE: Tests

ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.2 Security-enforcing functional specification
ATE_FUN.1 Functional testing

Objectives

409 The objective of this component is to confirm that all of the TSFIs have been tested.

Application notes

410 In this component the developer confirms that tests in the test documentation correspond to all of the TSFIs in the functional specification. This can be achieved by a statement of correspondence, perhaps using a table, but the developer also provides an analysis of the test coverage.

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_DPT.1 Testing: basic design

Dependencies: ADV_ARC.1 Security architecture description
ADV_TDS.2 Architectural design
ATE_FUN.1 Functional testing

Objectives

420 The subsystem descriptions of the TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the TOE subsystems provides assurance that the TSF subsystems behave and interact as described in the TOE design and the security architecture description.

Developer action elements:

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation elements:

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_FUN.1 Functional testing

Dependencies: ATE_COV.1 Evidence of coverage

Objectives

430 The objective is for the developer to demonstrate that the tests in the test documentation are performed and documented correctly.

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.2 Security-enforcing functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures
ATE_COV.1 Evidence of coverage
ATE_FUN.1 Functional testing

Objectives

444 In this component, the objective is to demonstrate that the TOE operates in accordance with its design representations and guidance documents. Evaluator testing confirms that the developer performed some tests of some interfaces in the functional specification.

Application notes

445 The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc.

446 This component contains a requirement that the evaluator has available test results from the developer to supplement the programme of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

AVA: Vulnerability analysis

AVA_VAN.2 Vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description
ADV_FSP.2 Security-enforcing functional specification
ADV_TDS.1 Basic design
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Objectives

460 A vulnerability analysis is performed by the evaluator to ascertain the presence of potential vulnerabilities.

461 The evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. Penetration testing is performed by the evaluator assuming an attack potential of Basic.

Developer action elements:

AVA_VAN.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.2.1C The TOE shall be suitable for testing.

Evaluator action elements:

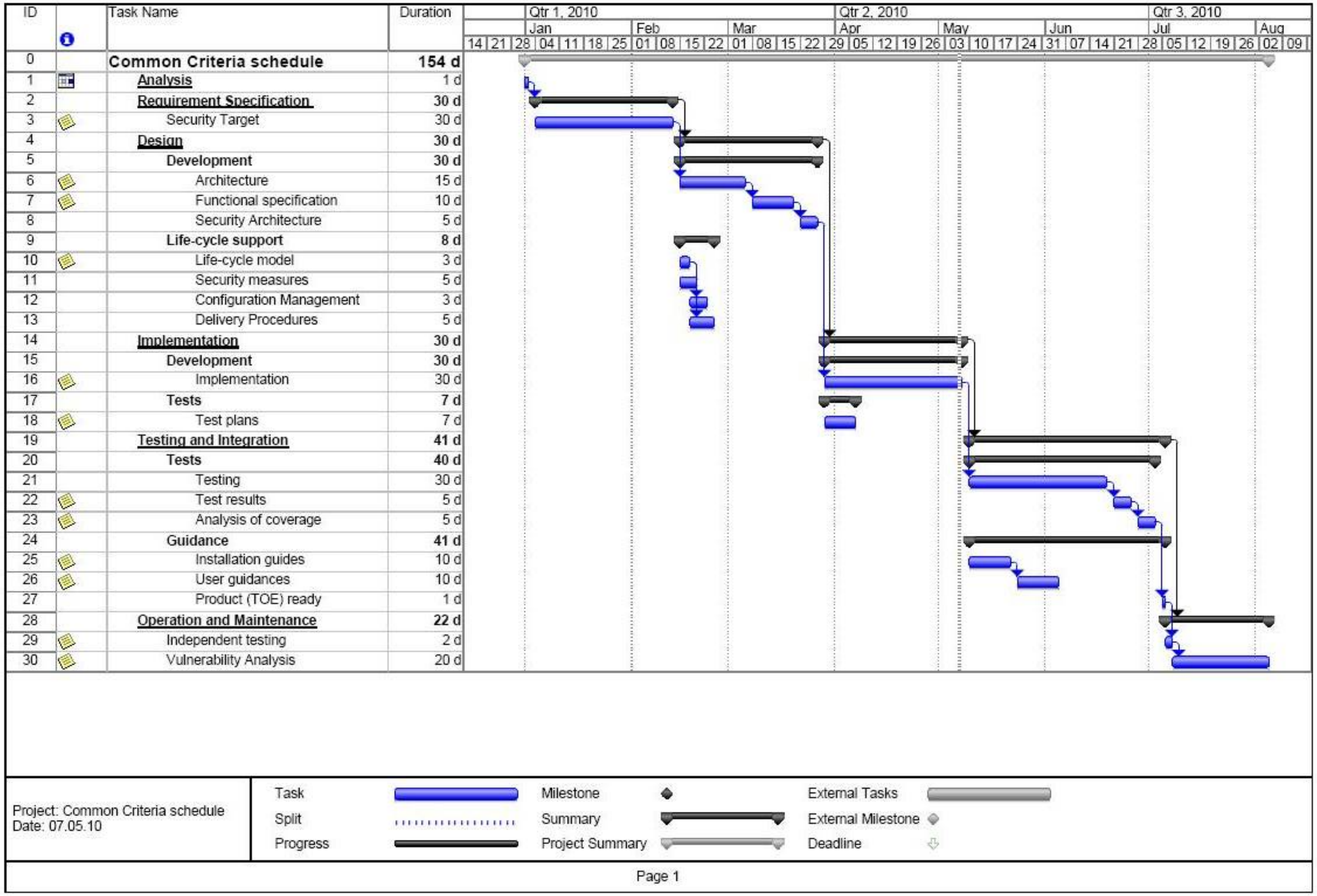
AVA_VAN.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator *shall perform* an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Appendix 2. Common Criteria Process Schedule



ID		Task Name	Duration
0		**Common Criteria schedule**	**154 d**
1		**Analysis**	1 d
2		**Requirement Specification**	**30 d**
3		Security Target	30 d
4		**Design**	**30 d**
5		**Development**	**30 d**
6		Architecture	15 d
7		Functional specification	10 d
8		Security Architecture	5 d
9		**Life-cycle support**	**8 d**
10		Life-cycle model	3 d
11		Security measures	5 d
12		Configuration Management	3 d
13		Delivery Procedures	5 d
14		**Implementation**	**30 d**
15		**Development**	**30 d**
16		Implementation	30 d
17		**Tests**	**7 d**
18		Test plans	7 d
19		**Testing and Integration**	**41 d**
20		**Tests**	**40 d**
21		Testing	30 d
22		Test results	5 d
23		Analysis of coverage	5 d
24		**Guidance**	**41 d**
25		Installation guides	10 d
26		User guidances	10 d
27		Product (TOE) ready	1 d
28		**Operation and Maintenance**	**22 d**
29		Independent testing	2 d
30		Vulnerability Analysis	20 d

Project: Common Criteria schedule
Date: 07.05.10