FOI
SWEDISH DEFENCE
RESEARCH AGENCY

FOI-R--0435--SE

March 2002

ISSN 1650-1942

Methodology report

Lars Westerdahl

Firewall Evaluation Criteria

- An evaluation

Issuing organization
FOI – Swedish Defence Research Agency
Command and Control Warfare Technology
P.O. Box 1165
SE-581 11 Linköping

Report number, ISRN
FOI-R--0435--SE

Report type
Methodology report

Research area code
4. C4ISR

Month year
March 2002

Project no.
E7023

Customers code
1. Research for the Government

Sub area code
41 C4I

Author/s (editor/s)
Lars Westerdahl

Project manager
Alf Bengtsson

Approved by

Sponsoring agency

Scientifically and technically responsible
Jonas Hallberg

Report title

Firewall Evaluation Criteria – An evaluation

Abstract (not more than 200 words)

In a time where more business is conducted over the Internet, the security of such systems, for instance electronic money transfer, becomes critical. A firewall serves as a frontline of the security measures that can be taken by an organisation in order to secure the integrity of a private network.

This report describes methods of evaluating the security of a firewall. The existing methods on the market are presented and an evaluation of these methods is performed using a reference model, developed for this thesis.

Two main categories of evaluation schemes are defined. Government schemes, that can produce a thorough evaluation and commercial schemes that are faster but does not offer such a rigor evaluation.

As a tool for comparison, a framework is presented. The framework is an abstraction of security evaluation that preserves the necessary properties needed in order to gain acceptance.

The master thesis is the result of study conducted at the Swedish Defence Research Agency in Linköping.

Keywords
Evaluation, Firewall, Security

Further bibliographic information
Master thesis Växjö University

Language English

ISSN 1650-1942

Pages 59 p.

Price acc. to pricelist

Security classification

Utgivare
Totalförsvarets Forskningsinstitut - FOI
Ledningssystemteknik
Box 1165
581 11 Linköping

Rapportnummer, ISRN
FOI-R--0435--SE

Klassificering
Metodrapport

Forskningsområde
4. Spaning och ledning

Månad, år
Mars 2002

Projektnummer
E7023

Verksamhetsgren
1. Forskning för regeringens behov

Delområde
41 Ledning med samband och telekom och IT-system

Författare/redaktör
Lars Westerdahl

Projektledare
Alf Bengtsson

Godkänd av

Uppdragsgivare/kundbeteckning

Tekniskt och/eller vetenskapligt ansvarig
Jonas Hallberg

Rapportens titel (i översättning)
Utvärderingskriterier för brandväggar

Sammanfattning (högst 200 ord)

I en tid då affärer over Internet blir allt vanligare, ökar säkerhetskraven på de system som hanterar transaktionerna. En brandvägg är ofta det första steget i en kedja av skyddsmekanismer avsedda att skydda ett privat nätverk.

Den här rapporten beskriver metoder för att utvärdera säkerheten i en brandvägg. De existerande metoderna på marknaden presenteras och utvärderas med hjälp av ett ramverk, speciellt framtagen för denna uppsats.

Två huvudgrupper av utvärderingsmetoder definieras. Metoder med statlig översyn och internationellt samarbete, vilka kan producera en rigorös utvärdering samt kommersiella metoder vilka är snabbare, men inte lika grundliga.

För att kunna göra en utvärdering presenteras ett ramverk. Själva ramverket är en abstraktion av en säkerhetsutvärdering och det bevarar de nödvändiga egenskaperna en säkerhetsutvärdering behöver för att accepteras.

Denna magisteruppsats är resultatet av ett examensarbete utfört på Totalförsvarets forskningsinstitut i Linköping.

Nyckelord
Utvärdering, Brandvägg, Säkerhet

Övriga bibliografiska uppgifter
Magisteruppsats Växjö Universitet

Språk Engelska

ISSN 1650-1942

Antal sidor: 59 s.

Distribution enligt missiv

Pris: Enligt prislista

Sekretess

Abstract

In a time where more business is conducted over the Internet, the security of such systems, for instance electronic money transfer, becomes critical. A firewall serves as a frontline of the security measures that can be taken by an organisation in order to secure the integrity of a private network.

This report describes methods of evaluating the security of a firewall. The existing methods on the market are presented and an evaluation of these methods is performed using a reference model, developed for this thesis.

Two main categories of evaluation schemes are defined. Government schemes, that can produce a thorough evaluation and commercial schemes that are faster but does not offer such a rigor evaluation.

As a tool for comparison, a framework is presented. The framework is an abstraction of security evaluation that preserves the necessary properties needed in order to gain acceptance.

The master thesis is the result of study conducted at the Swedish Defence Research Agency in Linköping.

Preface

This report is the final stage of the computer science program at Växjö University. It is a 20p (full semester) master thesis conducted at the Swedish Defence Research Agency in Linköping during the spring of 2001.

The report describes ways of measuring the security level of firewalls, as well as conducts an evaluation of known evaluation schemes.

A special thanks to my tutors;

- Ulf Cederling, Växjö University
- Jonas Hallberg, Swedish Defence Research Agency

for their support during the completion of this report.

Växjö 24 September 2001

Lars Westerdahl

Contents

List of tables

List of figures

1 Introduction

In a time where the Internet becomes a more natural part of the day-to-day routines for companies and organisations as well as private persons, the security of the information is essential. If an Internet bank cannot assure their users that the banks security system is sufficient, the customers will choose another bank which can. A company situated in different cities may want to be able to communicate with the different parts of the company in a confidential manner so that no business critical information is revealed. Even though the importance of the information receding in a computer or is sent over the Internet may vary, the integrity of it is just as important from a government point-of-view as well as from a private user point-of-view.

Information is not the only thing that needs to be protected. With the growth of networks, both site-networks and metropolitan-networks, which provides continuos high-speed online access, the resources of such networks become of interest for an attacker. By using several computers the attacker can, for instance, perform a more effective distributed denial-of-service attack or increase his computational power in order to break passwords.

Security can be provided in different manners, all depending on the value of what is to be protected and how much the owner of the information/resources is willing to contribute in man-hours and money. Unfortunately, there is not one solution that eliminates all security issues. A firewall can serve as the first line of defence in a chain of different methods, all contributing to the overall security standard of the network. If a firewall is set-up correctly, all in- and outbound traffic should traverse through the firewall, resulting in a unique ability to control the traffic. It is important, though, to realise that even if the traffic is controlled, legitimate traffic may also contain hazards. A firewall inspects packages, not the complete message sent through it. Other mechanisms such as antivirus program and intrusion detection systems (IDS) are used to enhance the overall security beyond the firewalls' capacity.

Estimating the efficiency of a security product, such as a firewall, is in many ways a difficult task. It demands a great amount of knowledge, time and resources, something that may be hard for a procurer to provide and impossible for a private person. For this reason third party evaluation is necessary. It provides an impartial evaluation of a product that, in turn, helps procures and private persons to choose a product suitable for their needs.

This report will explore the methods of measuring the security of firewalls. The methods presented differ in ambition, rigor and result. Some, for instance Common Criteria, are general concepts with adaptations to specific equipment, whereas others, such as ICSA FWPD Product Certification Criteria, are special purpose evaluation criteria.

A general model for security evaluation will be presented in the report. The model is not a functional security evaluation scheme, such as those presented, but a reference tool used to compare and evaluate the presented methods.

1.1 Problem area

Performing security evaluations is about confirming the known and predicting the unknown. The known part is the sum of the experience of the technology. Firewall developers and evaluators are no fortune-tellers in the sense that they cannot prove that a firewall will hold for any given type of attack. The future can only be guessed, but by confirming the rigor of the product some future problems may be avoided.

From an attacker's point of view, the method of operation is becoming simpler. Easy-to-use tools have been developed and spread over the Internet, allowing for less skilled people to perform devastating attacks.

National and international efforts have been made since the early 1970ies to construct security evaluation schemes. They differ in how rigor the evaluation is, how the result is presented, and the level of acceptance the evaluation yields.

1.2 Purpose

There are a few different security evaluation schemes on the market today. They differ in how they perform the evaluation and the recognition they receive. The purpose and the main issue of this thesis are to answer the question: ***Are the current methods of evaluation sufficient to evaluate the security of a firewall?***

In order to answer this question a framework will be presented. The framework functions as an abstraction of security evaluation thus making it possible to compare existing schemes that may vary in method and how the result of the evaluation is presented.

The purpose of the framework is to help answering sub-questions, all leading to the answer of the main question. The framework will consider the prerequisite of the evaluation, the methods used during the evaluation and which manner the result is presented.

- Prerequisite: What type of documentation the developer needs to supply to the evaluators and the assumptions made by the evaluators.
- Methods: How the evaluation is conducted.
- Result: In what manner the result presented and the duration of the result.

In firewall security evaluation the key issues are to confirm the correctness and the efficiency of the firewall. As far as correctness is concerned, it is about confirming that the planning and implementation of the firewall is equivalent with what the developer claims. Efficiency is about estimating and measuring how well the firewall performs its duties.

The methods used to evaluate the correctness and efficiency of the firewall determines both the documentation needed to be supplied and the way the result will be presented.

1.3 Methodology and demarcation

As this is an evaluation of methods this thesis is strictly a literature study. The evaluation is verified by the assumptions made in the framework. Most of the material used in the thesis can be found on the Internet.

Several of the evaluation schemes presented are written in general terms, not specifically aimed for a certain type of hardware or software. Others are specifically constructed for dedicated equipment. To provide consistency and to limit the complexity of the task, this report is limited to the study of evaluation schemes that deals with firewalls. Throughout the thesis a firewall is defined as a component or a set of components that restricts access between a protected network and the Internet, or between other sets of networks [CCZ00].

Though it is not a technical limitation, only the situation where the firewall is situated between a private network and the Internet will be discussed. A firewall residing between a single computer and the Internet shares the same properties as the definition, whereas a firewall situated between two segments of a private network has a simpler situation due to that all addresses are known. The reason for this limitation is to avoid repetition in the text.

An assumption made by all the security evaluation schemes as well as the presented framework is that the firewall is physically safe, i.e. only authorised persons can gain physical access to the firewall.

The physical shape of the firewall may vary. It is not necessarily that it is a stand-alone hardware item with a controlling software installed, it can just as well be a software program installed in a general purpose computer used for other duties as well. It is the software of the firewall that determines the functionality of the firewall, hence, it is only the software of the firewall that is taken under consideration.

An introduction to firewall design is presented in appendix I and a brief description of the TCP/IP communication protocols is presented in appendix II.

1.4 Disposition

The thesis is divided into three parts. The first part describes in general terms the concepts of assurance and how it is produced. This part is mostly based on position papers to the Information-Security-System Rating and Ranking workshop held in May 21-23, 2001, Williamsburg, Virginia [ISSRR].

Part two is a presentation of current and, in the case of the Trusted Computer System Evaluation Criteria, historical evaluation schemes. These are not the only security evaluation schemes on the market. I have chosen to address those most often referred to by other authors and that conduct technical security evaluation. The evaluation schemes presented are:

- The Trusted Computer System Evaluation Criteria [TCSEC]
- Information Technology Security Evaluation Criteria [ITSEC]
- Common Criteria for Information Technology Security Evaluation [CC]
- ICSA FWPD Product Certification Criteria [TRUE]
- Firewall Checkmark Criteria [FCC]

In the last part a general model for security evaluation is presented and used as a reference tool when comparing the evaluation schemes from part two.

Appendix A is a short introduction to firewall technology.

Appendix B is a brief description of TCP, UDP and ICMP.

2 Background

To provide trust in a product, some kind of evidence has to be established. This can be done by estimating the efficiency and correctness of the security enforcing functions or by testing the whole product. Usually both methods are being used to ensure customers that the product is suitable for their purposes. This chapter will present the possibilities of estimating the security characteristics of a product from a scientific point of view, and how to test it. Initially, this chapter will present the security problem in general terms, then see how it apply on a firewall. Later assurance, a common term when it comes to computer security, and how to achieve assurance are addressed.

2.1 The security problem

Computer security is founded on three pillars; availability, confidentiality and integrity [PFL97, 1]. The content of this concept varies depending on which part of the computer security area is being studied, for instance intrusion detection systems. A general description of the terms can be:

- Availability: The asset should be available to the ones who are authorised to use it.
- Confidentiality: Closely related to availability. It means that the asset is only accessible by authorised parties. Confidentiality is about keeping something secret, whereas availability is how to enable the right person to use the asset.
- Integrity: Only authorised parties should be able to modify the asset.

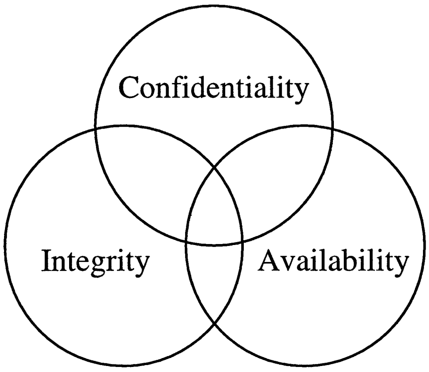

Figure 2.1. *The three pillars of security* [PFL97, 1].

Before modifying the concept to fit in a firewall perspective, a definition of a firewall is suitable. There are a few different opinions of what is to be defined as a firewall, though in this report I shall comply with Chapman, Cooper and Zwickys' [CCZ00] opinion. They state that a firewall is a component or a set of components that restricts access between a protected network and the Internet, or between other sets of networks.

As Chapman et. al concludes, a firewall can be installed between any kind of networks or segments of a network. The location of the firewall determines how it is going to fulfil its purpose. For instance, posit a firewall situated between two segments of a protected network. All machines on both segments are known, which makes it fairly easy to allow/deny access between the segments. The situation is more difficult when the firewall is situated between the protected network and the Internet. Nothing is known about the machines or users of the Internet, hence making it harder to determine what should be allowed and what should not.

To avoid repetition of which situation is referred to only the situation where the firewall is situated between a private network and the Internet will be described. This is not a limitation, just a simplification of the writing, as mentioned in the section 1.3.

Having a firewall installed between the protected network and the Internet, should create a bottleneck, i.e. all traffic should pass through the firewall. This results in the firewall being the first line of defence and gives the firewall the ability to control all the traffic, thus being able to enforce the rules set to protect the internal network.

These rules are a part of a larger plan for the overall security of the network, known as a security policy. In order for the firewall to work properly, it is necessary that all the users of the network comply with the security policy. It would be a major breach of security if, for instance a user decides to install a modem connection, thus allowing for unprotected access to the network. A firewall can only enforce the rules on the traffic that goes through it; it cannot protect the network from internal security issues, e.g. misbehaving users.

Putting the properties of the security problem in a firewall perspective they can be expressed like:

- Availability: Users of the internal network can communicate through the firewall in a way that complies with the network security policy. The administrators shall be able to perform maintenance.
- Confidentiality: Unauthorised parties are not allowed to access the internal network from the outside. The implemented security policy should regulate who has the right to access the internal network and who is authorised to reconfigure the firewall.
- Integrity: Only authorised parties are allowed to configure or audit the firewalls properties. There should not be any other way to access the firewall than through proper channels.

A good security evaluation scheme should confirm these properties.

2.1.1 Threats

The purpose of launching an attack towards a network may vary depending on the owner, the content or the resources [CCZ00, 1]. Example of purposes may be:

- Owner: Purpose of blemish the reputation of the owner or gaining commercial benefits.
- Content: Confidential or business critical information can be used or sold.
- Resources: An attacker may not be interested in who the owner is or what he does. The main purpose can be to gain computing power or disk space.

With the purpose clear, the next step for an attacker is to determine the goal of the attack, and a suitable method to achieve the goal. Either way, the first obstacle an attacker encounters is the firewall. How the firewall is dealt with depends on the purpose, goal and method of the attack. A small example may be appropriate.

GRC.COM where the target for a Denial-of-Service (DoS) attack on may 4th this year [GIB01]. The reason for the attack was revenge for claimed foul slander. A vast amount of UDP and ICMP packages where sent from 474 different computers to the company router, resulting in an overload, thus putting the company temporary out of business.

Purpose:	Revenge (attack against the owner).
Goal:	Hinder them to conduct their business.
Method:	Denial-of-service attack.

A denial-of-service attack is a type of attack, which is maybe the simplest to perform. Most often the attacker uses existing tools and has no other purpose than self-fulfilment. Other types of attacks may be to gain access to a piece of information or to use some asset behind the firewall. Either way, he has to find a way of being able to traverse the firewall. A skilled attacker will do so without leaving any compromising trails.

An attack aiming to access the network behind the firewall is often time consuming. The attacker has to gain knowledge of the network and find flaws in the security system, or find an ignorant user who, unknowingly, can participate in the attack.

These are technical threats. Other possible threats to a network are beyond the firewall capacity to control, such as insiders and other connections to the Internet. What happens behind the firewall is not of concern to the firewall, and therefore not dealt with in this report.

2.2 Assurance

Security is not only a matter of building products that perform a specific duty. It is essential that the product perform its task well. How can we be assured that the product behaves as intended?

No developer gives any guarantee that, for instance, a firewall can stop arbitrary attacks, or for that matter any guarantee at all apart from normal functional properties. Instead they try to assure a purchaser that their product will perform as intended.

Assurance is not the same thing as a guarantee. No compensation is given, if the product should fail in pursuing its objectives. Assurance is better described as a level of trust assigned to the product. As a purchaser, one has to define what it is one want to be assured of.

Whether it is a commercial, government or private user, taking protective measurements means that there is an interest in protecting something. It can be business plans, secret or sensitive material or simply an effort of conserving the integrity of the system or user.

Whatever the reason may be, there is a breaking point between efforts and costs. As an owner of a system, one must decide the effort one is prepared to devote in order to preserve the purposes. It is a question of time and money. Even if cost is no object, it is time-consuming to provide evidence of the systems' or products' safety.

What is it then that the system needs to be secure from? Again, it depends on the purpose of the system. Electronic commerce needs to provide safe communication and store sensitive information, such as credit card numbers. A lost or compromised connection can be disastrous for such an establishment. Not only will they loose valuable time, their reputation may decrease for every minute their system is unavailable. Hence, assurance that the system can be restored in a quick and safe manner is also of importance.

Government systems may contain information of national security, which needs to be kept secret for a long time. In such a situation, other aspects of security are of interest, compared with a commercial facility.

As a private user of a computer system, integrity is generally of prime concern. The information stored is most often of personal value rather than financial. Either way, no one should be able to use a computer without the owners' permission.

Whatever the reason for securing a computer or network is, some amount of time and money is needed. Probably the most common mean of protection is an anti-virus program. It is a fairly inexpensive and effective method of protection. The next step could be to install a firewall. These two approaches do not exclude one another, rarely any security enhancement does, they are more likely to complement each other.

A properly configured firewall should be able to prevent unwanted visitors to gain access to the system. As stated earlier, "*should*" is no guarantee. It is up to the developer or system administrator to provide facts in order to convince a purchaser or system owner that the designated level of assurance is achieved. Gathering these facts is the hard part. Various tests can be performed, such as trying to break into the system. Even if a commercial or government facility has the ability to hire an independent third-party team to perform testing of the system, it is hardly feasible to set up a random system and having it evaluated. Some criteria should exist when choosing the components in the first place. The same goes for a private user, who may not have the ability to test the system on his own.

Third-party evaluation is one way of getting a second opinion about a product or system. Especially for COTS (*C*ommercial *O*ff *T*he *S*helf - products you buy in a store) some kind of evaluation is recommendable, since it is hard to estimate, even for professionals, how well it will perform.

Assurance, though, is not a strictly technical problem. Even if a state-of-the-art system is installed, the users have to be aware of the risks involved when using the system. An email can pass through a firewall containing a malicious program, such as a Trojan horse. The mail itself is a valid packet, thus will not be blocked by the firewall. Not opening attached files without knowing the sender or the content is the responsibility of the receiver. To make users aware of the risks and how the owner of the network wants to deal with them, a security policy is established. It should state what is allowed and what is not. If the users know the security policy and why it exists, it is more likely that they will accept it. Not complying with the security policy, for instance installing a modem connection to access some service prohibited by the security policy is a serious breach of security. The modem connection will constitute a covert channel that can be exploited without the system administrators knowing anything about it. The details of a security policy are beyond the scope of this report, but it will be addressed when needed for explanation.

Another part of assurance is to know the limitations of the system. It is just as important to know the limitations as to know the strength of the system. In knowing the weaknesses, the administrators can compensate for them in other ways or at least have a backup plan if they are explored.

In general, assurance is established by technical equipment, rules of conduct and the people involved using and maintaining the equipment (figure 2.2). The equipment help to enforce the security policy through automated surveillance and the technical staff maintain and update the equipment. The users should be educated and kept informed about current events. Therefore assurance is something to improve and maintain over time, not a one-time solution.

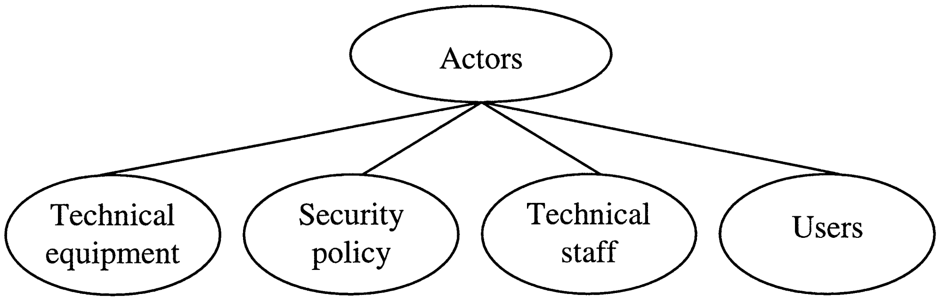

Figure 2.2. *Actors in establishing assurance.*

2.3 How to produce assurance

In order to convince a customer or a system owner of a components' assurance level, some kind of instrument is needed. One way is to try to estimate the protective values of the system by measuring how well the security enforcing functions perform. Another way is to test them, i.e. try to make them fail.

To provide assurance, some facts have to be established. These facts are provided by some kind of measurement. A measurement can provide quantifiable facts about length, width, amount, buffer capacity and so forth. Mathematics can be used on measurements to generate area, volume and safety marginal. This is objective information. Measurement by itself does not provide any useful information other than for direct comparison with similar products [ALG01].

To make measurements meaningful and to make them possible to use as a decision-making tool, they should be turned into metrics. A metric is a measurement put in reference to something. Let's clarify this by an example.

> Passwords that are very short, say two to four characters, provide low security since it does not take very long to break them with common password breaking tools. By setting up a subjective goal, for instance that it should take a minimum of *x* minutes to brake a password, and testing the stamina of different lengths of passwords we achieve a result both relying on objectivity and subjectivity. Time is the objective part and the measurement, whereas the statement of acceptable time is a subjective call. The resulting metric is a test of efficiency regarding length of password. The knowledge can be used when

writing a security policy or by implementing a function that does not allow passwords shorter than a predefined number of characters [BAR01].

Another positive property of metrics is that the scale used does not have to meet any general standard and, hence, can be adjusted to fit the purpose.

As a decision-making tool, this comes in handy. A system can be evaluated to meet a security policy – a policy that usually does not translate well into standardised measurements. This is not to say that metrics should be used to prove something arbitrary. A metric still has to be as rigid as a measurement. It should follow the rules of basic science, such as reliability, validity, and reproducibility et cetera listed by Maxion [MAX01].

Later, when comparing existing evaluation methods, the following variables will be taken under consideration:

- Validity: Have we measured what we where supposed to.
- Reproducibility: Can it be done again with the same result?
- Control: Are we aware of all of all the variables that contribute to the result?
- Terminology: Can we express our inputs and result in a commonly accepted manner?
- Ethics: Is the result achieved using accepted (read legal and moral) methods, and how is the result presented?

These variables are just a portion of what Maxion suggests. Validity is the basics for the result. It is a receipt showing that the correct measurement has been taken to achieve the wanted result. Control is important for reproducibility. If control is not achieved, than a new evaluation may yield a different result without changing any of the predefined conditions. The result would be an unreliable evaluation.

To avoid misunderstandings and to make it easier for all parts, a common language should be set. In doing so time can be saved both during evaluation and, if necessary, when the developer corrects the firewall.

Ethics is not so much about correctness and efficiency, as it is with acceptance. Security is a sensitive area of business and evaluators may be supplied with information that is not for public use. The integrity of the evaluation is critical. It is in the developer interest that the result is published, but that no security-sensitive information is included.

2.3.1 How to use metrics

Since the beginning of security evaluations, there has been a quest for a single measurement. Unfortunately, security is not a single object to install or a single operation to perform. The following example shows why it is so.

> Say that we have a got a product that has ten different functions and we wish to choose two of these. Then there are 45 different pairs of combinations and 1024 different configuration all together, each with their own security implications.

For a computer system, ten settings is a small number, hence the complexity of an evaluation is vast. Security flaws often occur when two, by themselves harmless, modules interact [PFL01].

Katzke [KAT01] suggests a general security metrics model composing of three pillars;

1. The object,
2. The security objectives, and
3. The method of measurement.

The object is what is going to be measured. It does not necessarily have to be a product or system; it can be a security policy or the competence of the staff. A set of security objectives is established, which should reflect the accurate security policy or being goals for the object to meet. The methods for measuring should be chosen on the basis of the goal of the test. This may include methods like evaluation, direct testing or assessment.

3 Methods of evaluation

The history of computer security evaluation is quite short. Officially recognised evaluation schemes only dates back to 1985, beginning with the Trusted Computer Security Evaluation Criteria (TCSEC). Even though TCSEC is outdated, it has served as an important base for other evaluation schemes.

Studying the evaluation schemes used today, two main directions can be found. There are those which are government controlled and those run by commercial organisations [HAHU99]. The government controlled schemes usually divide the responsibilities of performing the test on the systems and the act of certifying them. The responsibility of the government is to maintain the evaluation criteria and to approve test facilities.

These evaluation schemes are very thorough, requiring a vast amount of documentation of the developer to marshal his claims of the security capabilities. The evaluation methods range form validation to formal methods. Validation is the simpler form of confirming the developers' claims usually performed by requirement checking, design and code reviews, along with module and system testing. Formal methods are not necessarily more complex, but they are harder to prove. Typical formal verification divides the problem into smaller parts, theorems, which is to be proven, usually by the use of flowcharts [PFL97, 2]. Only the higher assurance levels of the evaluation schemes require formal verification. The original purpose of these evaluation schemes was to aid government procures in choosing the right components. Commercial components (COTS) have also been submitted for evaluation, which is a proof of its acceptance.

Being thorough is of course costly – both in time and money. Due to this, some commercial organisations have recognised the need for a more swift evaluation scheme. Their purpose is not to investigate every possible characteristics of the security problem, rather provide a basic level of assurance in a short period of time.

This is the reason why government evaluation schemes are much more complex and the amount of documentation needed is larger. Commercial evaluation uses a black-box approach, meaning they do not consider the internal solutions or design. It is only the result of their tests that determines whether or not the firewall will receive an approval. Government evaluation schemes take the design as well in consideration when performing the evaluation. To put it in simple terms; government evaluation schemes evaluate the correctness and the effectiveness of the firewall, whereas commercial evaluation schemes merely consider the effectiveness of the same.

The assurance requirements for government agencies and commercial organisations differ in the sensitivity of handled information. Government agencies, e.g. military and law enforcement need to protect information ranging from sensitive but not classified to top-secret. The hierarchy of secrecy is not always strict. "*Need to know*" is a variable that can allow a less privileged person to access specific, high sensitive, information.

Commercial organisations rarely handle classified information. Prime concerns of commercial organisations are to maintain their privacy in order to, for instance, conduct electronic commerce and to be cost-effective. The value of keeping information secret usually expire faster for commercial organisations then for government agencies.

This chapter will present the most accepted evaluation schemes – government and commercial. There are other evaluation schemes, such as ISO 17799 (previously known as British Standard 7799 (BS7799)) [ISO17799], but I will only address those that evaluate products.

3.1 Government evaluation schemes

By the end of the 1970s the Department of Defence (DoD) became aware of the difficulty for acquisition personnel to specify their needs or to evaluate their systems. They saw the need of an evaluation scheme that could function as a tool to support decisions when acquiring new computer products or to help evaluate the existing systems.

3.1.1 The Orange book

The Trusted Computer System Evaluation Criteria (TCSEC) [DOD85], or more popular the Orange book due to its colourful cover, was published in 1985, as a result of the combined effort by US Department of Defence and the MITRE Corporation.

The purpose of the criteria was to be able to provide users with a measurement of what a system could offer in terms of security. The same measurement can be used to specify security requirements in acquisition specifications. Further, it may also provide as guidance for future products.

The criteria is divided into four main divisions ranging from A to D:

D: Minimal protection.
C: Discretionary protection.
B: Mandatory protection
A: Verified design.

In division B and C different classes of security are added, resulting in the complete range of classification from (lowest to highest): D, C1, C2, B1, B2, B3 and A1.

Division D

This division constitutes of one class only. Systems that fail the requirement set in the evaluation for the intended level of certification belong to this class.

Division C

Class C1: Discretionary security protection

The class applies in an environment where users are able to share data in such way that they can control what to share and not. Functions for access limitation on an individual basis and authentication are needed. In a typical class C1 environment, all users share data of the same level(s) of sensitivity.

Class C2: Controlled access protection

The control level of class C2 increases somewhat, in logging user activity and forcing them to logon to a user profile. Thus it will be possible to track access or access attempts to an object down to a specific user.

Division B

Class B1: Labelled security protection

Each controlled subject or object must have the possibility to be assigned a security level. Access control decisions are made on the basis of the security level. Also, the access control must be based on a method supporting both hierarchical level and non-hierarchical categories. Example: Top Secret, Secret, Classified and Unclassified are levels of hierarchy, whereas Need-to-know is more of a floating category.

Class B2: Structured protection

The level of documentation of the formal security policy increases. All objects and subjects are to be labelled. Covert channel analysis is being done.

The resistance to penetration is relatively high.

Class B3: Security domains

The keyword for class B3 is simplicity. The security functions must be tamperproof and small enough for extensive analysis and tests.

The system is highly resistant to penetration.

Division A

Class A1: Verified design

The top class acquires a formally verified design of the system. The developer should provide evidence that the formal model of the protecting system is correct, and a formal top-level specification of the protection system.

Table 3.1 shows how the demands increase over the classes.

At the time TCSEC was developed computer networks were not in common use. By the time of publication, the use of computer systems had changed. The original TCSEC document was aimed for stand-alone computer systems. Due to this a library of interpretations has been produced to cover other systems and special purpose functions. The library contains documents regarding, for instance, networks (Trusted Network Interpretation (TNI) of the TCSEC) and databases (Trusted Database Interpretation (TDI) of the TCSEC). The library is also known as the rainbow series, also due to the colour of their covers.

Nothing new is being certified according to the TCSEC standard today. Due to its inflexibility, it was to be replaced with Federal Criteria (FC). The work of FC was never finished due to the development of the Common Criteria (see 3.1.3).

Even though the TCSEC is not an active document today, it is still interesting, as it served as a foundation to other certification models and is often referred to when discussing security evaluations.

Criteria	D	C1	C2	B1	B2	B3	A1
Security policy							
Discretionary access control	–	X	X	⇒	⇒	X	⇒
Object reuse	–	–	X	⇒	⇒	⇒	⇒
Labels	–	–	–	X	X	⇒	⇒
Label integrity	–	–	–	X	⇒	⇒	⇒
Exportation of labelled information	–	–	–	X	⇒	⇒	⇒
Labelling human-readable output	–	–	–	X	⇒	⇒	⇒
Mandatory access control	–	–	–	X	X	⇒	⇒
Subject sensitivity labels	–	–	–	–	X	⇒	⇒
Device labels	–	–	–	–	X	⇒	⇒
Accountability							
Identification and authentication	–	X	X	X	⇒	⇒	⇒
Audit	–	–	X	X	X	X	⇒
Trusted path	–	–	–	–	X	X	⇒
Assurance							
System architecture	–	X	X	X	X	X	⇒
System integrity	–	X	⇒	⇒	⇒	⇒	⇒
Security testing	–	X	X	X	X	X	X
Design specification and verification	–	–	–	X	X	X	X
Covert channel analysis	–	–	–	–	X	X	X
Trusted facility management	–	–	–	–	X	X	⇒
Configuration management	–	–	–	–	X	⇒	X
Trusted recovery	–	–	–	–	–	X	⇒
Trusted distribution	–	–	–	–	–	–	X
Documentation							
Security features user's guide	–	X	⇒	⇒	⇒	⇒	⇒
Trusted facility manual	–	X	X	X	X	X	⇒
Test documentation	–	X	⇒	⇒	X	⇒	X
Design documentation	–	X	⇒	X	X	X	X

Signs: –:No requirement; ⇒:Same as previous class; X: Additional requirement

Table 3.1. *Class requirements in TCSEC* [PFL97, 3].

3.1.2 ITSEC

By the end of the 1980s England, Germany and France independently produced evaluation criteria for information systems. Soon they decided, together with the Netherlands, to cooperate in order to produce a common evaluation scheme valid over the whole of Europe. The result is formally known as the Information Technology Security Evaluation Criteria, though the abbreviation ITSEC is more often used.

The ITSEC is a government run evaluation program. The government is responsible for setting the standard and maintaining it up-to-date. To keep the certification impartial, independent third parties perform the actual evaluation. These parties known as Commercial Licensed Evaluation Facilities (CLEFs), are licensed by the Certification Body.

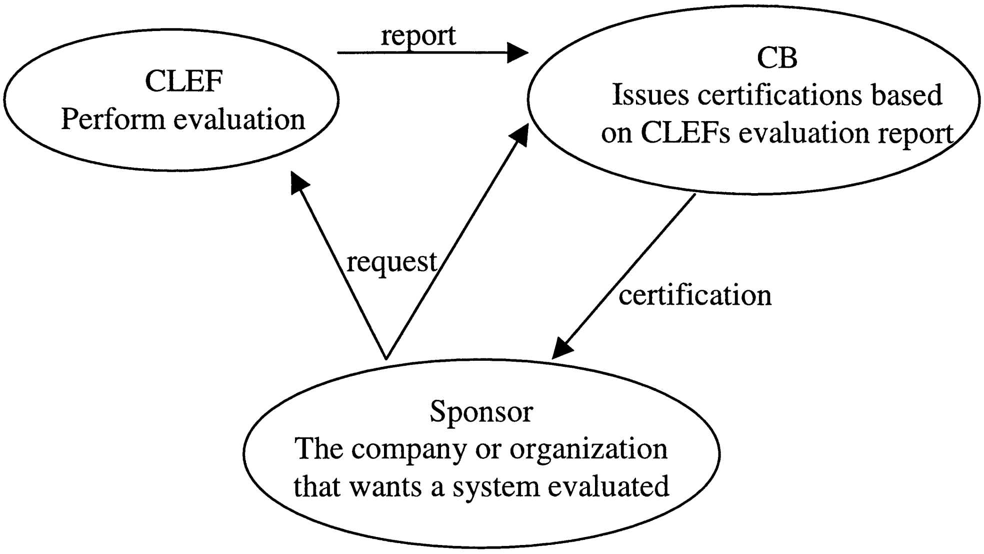
CLEF
Perform evaluation
report
CB
Issues certifications based on CLEFs evaluation report
request
certification
Sponsor
The company or organization that wants a system evaluated

Figure 3.1. *ITSEC organization.*

The Certification Body formally approves the evaluation report produced by the CLEF. They issue the certification for the system or product that has undertaken the evaluation.

In order to evaluate a system or a product (referred to as a Target Of Evaluation, or TOE), the sponsor has to define a security target, which constitutes the basis of the evaluation. It has to contain all the information necessary to understand the environment where the TOE is going to operate.

The level of assurance a certification yields can be divided into the confidence of the functionality and the effectiveness of the TOE. A correct TOE has the security functions correctly implemented, whereas the effectiveness regards the security of the TOE in reference to the security target (description of the TOEs operational environment).

An ITSEC evaluation will result in a level of assurance. A TOE is not submitted for evaluation with the intention of seeing how good it is. The evaluation result is binary in the sense that either the TOE passes the intended level or it fails.

The scheme has seven levels ranging from E0 to E6. The lowest level, E0, has no requirements. As with TCSEC, it is reserved for failed attempts. The documentation needed to evaluate a TOE increases with the level of assurance. As said earlier, the process differs between correctness and effectiveness. It is the correctness documentation that needs to be more rigorous with a higher level, the effectiveness criteria does not change.

The correctness documentation is summarised in table 3.2.

As seen in the table, there is a lot of documentation needed from the first level already, though the largest increase is between level E3 and E4. Here, the level of formality increases in the requirement, architecture and design as well as in implementation details.

The correctness documentation is evidence that the TOE is performing the duties the developer claims it to do, whereas the effectiveness evaluation actually tells you if the TOE is good at it.

Level	Documentation	Information of use to operational vulnerability analysis
E1–6	Security target	Definitions of threats, security objectives, intended environment, method of use, SEFs and mechanisms.
E1–6	Architectural design	TOE interfaces, interactions between functions, supporting hardware/firmware/software, and design of SEFs and mechanisms.
E2–6	Detailed design	Internal structure and interactions between components, design of SEFs and mechanisms and components.
E4–6	Source code and hardware drawings	Implementation details.
E5–6	Object code	Run time information.
E1–6	Operation documentation	Installation of TOE, configuration and use of security features including secure start-up, handling of security events such as warnings and alarms, diagnostics and error handling.

Table 3.2. *Documentation demands for different levels* [ITSEC, 2].

The effectiveness part of the evaluation is concerned with the practical protection of the TOE. ITSEC summarise effectiveness as "are the security measures implemented in the system or product effective against the threats identified in the security target and free from exploitable vulnerabilities?"[ITSEC, 1].

A higher levels of evaluation demands more comprehensive correctness documentation, but the effectiveness documentation remains the same. This is not to say that a level E1 TOE is as effective as a level E6 TOE. As the amount of documentation increases, so does the possibility of a more thorough evaluation.

An effectiveness evaluation is divided into six parts:

- Suitability analysis,
- Binding analysis,
- Strength of mechanisms analysis,
- Construction vulnerability analysis,
- Operational vulnerability analysis, and
- Ease of use analysis.

The aim is to provide assurance that each mechanism as well as the whole TOE provides a safe use, i.e. is free from vulnerabilities.

The suitability analysis determines whether or not the security enforcing functions (SEFs) are suitable to protect against the threats they aim to according to the security profile. It should also show if any threat has been neglected or improperly addressed.

Whereas the suitability analysis focuses on each SEF independently, the binding analysis tests how the SEFs of a TOE work together as a whole. Even if each SEF behaves excellent when tested separately, they might contradict or bypass each other when put together. It is the task of the binding analysis to find such vulnerabilities.

The main weapon for testing strength is to perform direct attacks. The evaluation team tries to penetrate the mechanisms in order to establish whether the claimed strength is valid.

They specifically look for vulnerabilities in implementation in order to determine the overall strength. Apart from attacks by a simulated intruder, insecure operations are performed as authorised users. Insecure operations are those a user can perform, which may result in an insecure state for the TOE, i.e. a situation where the ease of use compromise the security of the TOE.

Construction and operational vulnerability analysis are actually two separate analyses. However, since they are very similar, they will be described together.

The construction vulnerability analysis aim to confirm that the developer has addressed the vulnerabilities in the security target. The developer needs to provide evidence of how a vulnerability is met and that it cannot be exploited. As a follow-up the operational vulnerability analysis tests the TOE for known vulnerabilities.

The ease of use analysis works as an extension of the vulnerability analysis. The primary target is human activities. The test should determine if a TOE could be installed, configured or operated in an insecure manner, yet letting the user believe it is secure.

An important part considered is the aspects of failure. The security of the TOE or its objectives is not allowed to be any less secure due to failures. It should be obvious for a user if a hardware failure or a power supply interruption has occurred and if there are any consequences in regard of this.

The security target

For an evaluation to be meaningful, it has to be put in perspective of something, for instance, what it is supposed to protect and what is it protecting from. The security target provides this perspective by specifying the security functionality of the TOE and describes the environment, which it is intended to operate in. This constitutes the basis for the evaluation.

A security target should provide:

- A system security policy or a product rationale.
- A specification of the required security enforcing functions.
- The claimed rating of the minimum strength of mechanisms.
- The target assurance level.

Optionally, a definition of the required security mechanisms may be included in the evaluation. There are various reasons why a sponsor should include the definition of security mechanisms. He might do it for commercial reasons, to be able to provide a well-known edge in the advertisement. It is also good to add a specification if the sponsor is mandated to use a specific algorithm for a system.

A TOE can be a complete system or a product (COTS). In a system, the environment is known and shall be described in a system security policy. For a product though, the specific environment might not be known. A product rationale is description of the demands of the environment, in order for the TOE to be able to provide the stated assurance.

In a known environment, rules and procedures can be applied to ensure safe operation of a system. A system security policy states the preference of such a system. Hardware configuration and interfaces towards other systems are defined. It should also contain a specification of the type of users and operational role for the intended system.

On the basis of the defined system, security objectives and the threats towards them are identified along with the countermeasures designed to avoid or counter these threats.

It is harder to estimate the environment for a product, since a developer cannot force a customer to use it in a specific environment. A product may be a stand-alone entity or a component used in a system. Hence, the developer has to provide a description of the type of environment where the product can operate safely. The description is referred to as a product rationale.

A product rational can include technical aspects such as hardware configuration and needed software, but also aspect of the physical environment where it is going to be located. A detailed description of intended method of use is necessary to explain the purpose of the product and its features.

If the product has any dependencies with other hardware, software or firmware it should be stated in the document, as well as an estimation of the threats towards the security of the product within the described environment.

The security objectives are the highest level of security requirements included in the system security policy. It is not mandatory for a product rationale, but recommended. In a system, assets needing protection are defined. For a product, assets that the product is intended to protect are described.

A threat is defined by ITSEC as "*an action, which might prejudice security*". They are described in a more detailed way than the security objects. A security object should face at least one threat.

The precise description of a threat includes the source of the threat, for instance, a denial of service attack, and what the breach of security is, i.e. how the TOE is compromised. It shall also state how the breach is achieved.

The features of the TOE acting to protect the security objectives are referred to as security enforcing functions (SEFs). They are the main considerations of the evaluation, since they provide the security for the TOE. The developer who wants a product or system evaluated needs to be very specific about what the SEFs are doing and how they do it.

Functions that are not primarily security enforcing but in some manner supports the SEFs should also be addressed. It could be a hard disk containing passwords or user sensitive information. These are called security relevant functions and they are as much part of the evaluation as the SEFs.

The sponsor has to claim the level of strength the TOE can maintain. The levels basic, medium and high represent the amount of knowledge and resources an attacker should need to compromise a security mechanism.

Basic protection merely protects against accidental misuse, though it should not be possible for a layman to conduct a successful attack in a short period of time. It does not possess any major obstacle for a knowledgeable attacker, though. If the mechanism can withstand an attack from a skilled attacker with limited opportunity and resources, the claim can be set as medium.

With a claim of high level of strength, only a highly skilled attacker with lots of resources and opportunity should be able to defeat the mechanism.

In this case, resources mean the amount of time at hand, if there is inside help, for instance a user or a system administrator, and if special equipment is needed.

The overall level of strength is concluded, comparing all individual mechanisms and possible attacks. In order to claim a certain level of strength, there must be at least on mechanism with the same, or greater, level of strength along the way to a security objective. In other words, the layer must comply with the claims.

Finally, the sponsor should specify the evaluation level for the TOE. The levels indicate the amount of confidence one can put in the TOE, regarding correctness and effectiveness of the implementation of the security enforcing functions and mechanisms.

3.1.3 Common Criteria for Information Technology Security Evaluation

In the beginning of the 1990s a few different evaluation methods had been produced. They varied in acceptance and thoroughness, but the major problem for developer was the fact that an evaluation was not valid on another markets.

To meet the demand of world-wide acceptance, the authors of CTCPEC (Canadian criteria), FC, ITSEC and TCSEC gathered to form a mutual evaluation scheme, valid world-wide. The result was the Common Criteria for Information Technology Security Evaluation, more commonly known as Common Criteria.

The main resemblancė with previous methods is the incorporation of the protection profile from FC and the security target from ITSEC. A difference is that the lowest assurance class is not meant for failed evaluation, as it is in ITSEC. Instead it imposes a minimum level security class.

As with the other evaluation schemes, the Common Criteria is meant for all different kinds of IT equipment. Hence, the evaluation scheme is very general. The specifics come from the protection profile and the security target. An evaluation states whether a TOE meets the specifications of a security target, whereas a protection profile addresses the requirements for a TOE.

Common criteria defines seven evaluation assurance levels (EAL); EAL1 – EAL7. Level EAL2 – EAL7 corresponds directly to ITSEC levels E1 – E6. It is the first level, EAL1, which differ. In ITSEC the lowest level was reserved for TOEs failing an evaluation. In order to avoid failed products to use the evaluation methods name for commercial purposes ("this product has an E0 level of security"), the first level requires a minimum level of security. Failed TOEs are not given any class recognition. Another reason for the lowest class is to compete with commercial evaluation schemes. Usually an evaluation is a slow and costly process. By performing a minimum of assurance testing, such as basic penetration testing, the TOE can be assured that no obvious flaws of security are present.

EAL1	Functionally tested
EAL2	Structurally tested
EAL3	Methodically tested and checked
EAL4	Methodically designed, tested, and reviewed
EAL5	Semi formally designed and tested
EAL6	Semi formally verified design and tested
EAL7	Formally verified design and tested

Table 3.3. *Common criteria assurance levels.*

The general scheme is established by addressing security issues in terms of security requirements. The requirements sharing a common focus are gathered in general classes. Each class has a number of members, families, which share security objectives at different levels. The smallest part of the hierarchy is a component. A family includes one or more components. A component describes a specific set of security objectives.

A distinction is made whether it is a functional or assurance class. Functional classes, and its successors, describe how the security requirements are met in the design, implementation and operation of the TOE.

Functional classes	Assurance classes
Communication	Configuration management
Cryptographic support	Delivery and operation
Identification and authentication	Development
Privacy	Guidance documents
Protection of trusted security functions	Life cycle support
Resource utilisation	Protection Profile evaluation
Security audit	Security Target evaluation
Security management	Tests
TOE access	Vulnerabilities assessment
Trusted path/channels	
User data protection	

Table 3.4. *Classes in Common Criteria.*

Assurance classes works as guidance to both sponsors and evaluators. Each evaluation assurance level (EAL) contains a number of assurance components that has to be met in order to guarantee the level of confidence. An assurance component states exactly what a sponsor should provide in form of documents and how an evaluator should use these documents, as well as what individual tests that has to be performed.

A component can serve as a basis in the construction of a protection profile or defining a security target. A few components can also be gathered in packages if they form a mutual alliance towards meeting a specific subset of security objectives. The evaluation levels are predefined packages of assurance.

Protection profile

The protection profile, inherited from FC, is an evaluated assessment of the requirements needed for a general category of TOEs, e.g. a firewall, in order to perform a task safely.

It is not aimed for a specific developer or user thus can be reused as a guide or as the basis when developing new protection profiles or products.

A purchaser can write a protection profile as a specification of his needs, whereas a developer can use it as a description of his products' capabilities.

National Institute of Standards and Technology (NIST) and National Security Agency (NSA) has developed two protection profiles for firewalls, one for application level firewalls and one for traffic filter firewalls [NCSC99].

A protection profile contains a predefined amount of information (Table 3.5). Since it does not describe a specific product or, in some cases, a known environment, it contains several assumptions.

A few headers in the profile deserve some special attention. The TOE description provides the context of the evaluation. The description can be seen as an abstract for the system or product. It is used to identify inconsistencies during the evaluation.

The rationale summarises the whole profile. It explains how the chosen security objectives and requirements are met.

Protection profile

- PP Introduction
  - PP identification
  - PP overview
- TOE Description
- TOE security environment
  - Assumptions
  - Threats
  - Organizational security policies
- Security objectives
  - Security objectives for the TOE
  - Security objectives for the environment
- IT security requirements
  - Security requirements for the IT environment
  - TOE security requirements
    - TOE security functional requirements
    - TOE security assurance requirements
- PP application notes (optional)
- Rationale
  - Security objectives rationale
  - Security requirements rationale

Table 3.5. *Structure of a protection profile* [CC, 1].

3.2 Commercial evaluation schemes

The turnaround time for computer products is short. For companies that develop or change their equipment often, a time consuming evaluation may not be a feasible option. Commercial evaluation schemes offer a faster evaluation, assuring the buyer that the firewall can handle the most common types of existing threats.

3.2.1 ICSA FWPD Product Certification Criteria

The downside of rigorous evaluations like CC and ITSEC is money and time. Common software product has a relatively short lifecycle, and, during that cycle, several updates of various kinds are available. Performing a thorough and costly analysis over a year or two, only results in an evaluated product which version has been updated or replaced. In order to achieve consistency and maintaining assurance, the updated versions have to be evaluated too, resulting in a chain of evaluations.

TrueSecure (formerly ICSA) recognized this problem when launching their Firewall Product Developers' Consortium, or FWPD [HAHU99]. It is an evaluation scheme mainly aimed at testing a firewall's strength against penetration. Commercial products and companies rarely handle classified material of importance for the national security, their need is manly to prevent business and customer critical information from reaching into the wrong hands, and hence a comprehensive evaluation is not always needed.

From the developers' point of view, no additional documentation other then the ones delivered with the firewall is needed. The evaluators configure the firewall with a predefined security policy, using the same information a customer would receive.

The test phase is a black-box approach, which means that the internal design and implementation are of no concern. Penetration tests are mainly performed using commercial testing tools, such as ISSs Internet scanner and System Security Scanner. The purpose of the evaluation is to determine if the firewall can be configured and run in a safe manner using the customer information provided by the developer.

There are no levels of assurance, either a product passes or fails. A passed evaluation generates a certification lasting for a year. To allow for updates, random compliance checks are performed two to four times a year. Any changes in the evaluation criteria have to be met at each evaluation. A failed re-evaluation or compliance check results in a public announcement of revoked certification. The responsibility of maintaining a certification lies on the developer.

A certified firewall allows for FTP, HTTP, HTTPS, SMTP, DNS and TELNET (outbound only) traffic. All other traffic should be denied. To be able to track possible misuse, logging with timestamps, protocol type, source and destination should be presented in a readable fashion.

The security test proves that:

- Only authorised personnel can access the administrative mode.
- There are no vulnerabilities, known to the Internet community. Neither should the firewall present a threat to other servers.
- No other traffic than specified is able to traverse the firewall.
- It shall persist trivial denial of service attacks. If it should fail, caused by a more serious attack, it should fail/safe, i.e. deny all traffic.

The commercial tools used to test the security are regularly updated to be able to encounter newly discovered threats. A new vulnerability discovered is usually incorporated within 60 days.

3.2.2 Firewall Checkmark Criteria

The West Coast Labs offers a one-time certification for a product. Any updates have to be submitted for an evaluation.

The evaluation is conducted using a proprietary scanning tool, configured with full knowledge of the firewall and the network. The tests aim to establish a level of confidence in the firewall, by attempting penetration.

In many ways the Firewall Checkmark Criteria is similar to the FWPD Product Certification Criteria. They both aim to provide a quick evaluation that result in a certification stating that it should withstand common types of attacks.

While the Firewall Checkmark Criteria, like ITSEC and Common Criteria, only offers a one-time certification, ICSA provide more contingency by allowing for updates.

4 A framework for method evaluation

As stated earlier, there is no single measurement that can be performed to provide assurance. It is the combination of several measures that can offer some assurance. As an instrument of comparison, a general model for security evaluation will be introduced. This model will later serve as a framework when evaluating the methods presented in chapter three.

Recall Katzkes approach to define a security metric in chapter two. The framework presented in this thesis is inspired by his approach. The general idea of Katzkes model consists of three parts. There is an item to be evaluated, claims of what the item can do and a method to confirm that the claims are achieved.

By modifying Katzkes approach, an evaluation can be described as residing on three pillars; knowledge, method and result, shown in Table 4.1 and Figure 4.1.

Katzkes model for security metrics	Proposed model for security evaluation
Object The object that is to be measured, in this case a firewall.	**Knowledge** By combining Katzkes object and security objectives, the sum of what is known to the evaluators is gathered in the knowledge component.
Security objectives The demands of the firewall, i.e. what it is supposed to protect.	
Method of measurement The actual method that determine if the firewall can achieve the security objectives.	**Method** The guidelines the evaluators follow, i.e. the evaluation scheme.
Even if Katzke does not specifically mention anything about the result, the method of measurement should derive the properties of the result.	**Result** The way the result is presented. The evaluation scheme determines whether the result should be stated as a threshold or a scale. Either way, it should be motivated by stating the conditions and restrictions of the approval.

Table 4.1. *A comparison of Katzkes model and the proposed model.*

In his approach, Katzke is mainly concerned with providing a method of how to measure the security of a product. The framework in not only concerned with the “how”, but also the prerequisite of the evaluation and the outcome of the result.

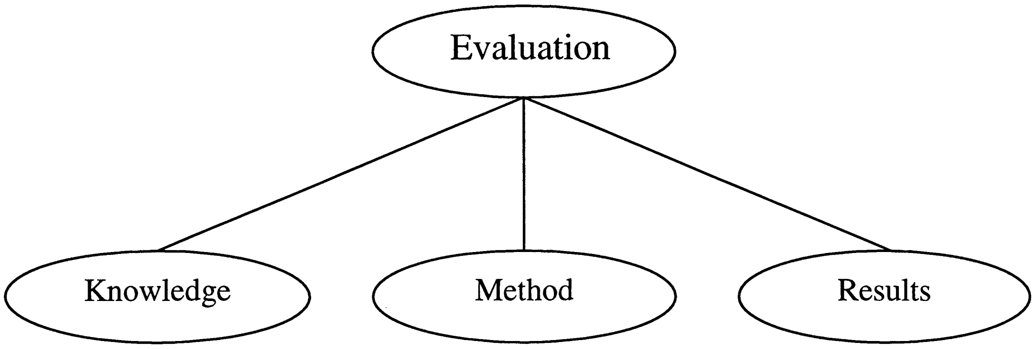

Figure 4.1. *Pillars of evaluation.*

The similarity with Katzke is in the generality. Knowledge is everything known about the object. Apart from the object itself, this means presented documentation, a knowledge bank of known vulnerabilities and the ways of exploring them. The method part is the same as with Katzke; it is a way of proving that the object can fulfil its goals.

For an evaluation to become successful, the integrity of the result is important. The utopia is that it should be impeccable, meaning that it should be valid forever. This would make it an absolute truth, which is impossible even in the best of worlds. If the result follows Maxions statement of scientific rules for dependable measurements it should suffice.

4.1 Knowledge

The first part of the framework is twofold. The developer provides one half of the knowledge and the other half is based on the competence of the evaluating facility.

What the developer should provide is determined by the security evaluation scheme. Basic requirements are of course the object of evaluation, the guidance documentation that comes with it and the security objectives that the object should fulfil. For the security objectives to make any sense a description of the intended operational environment should also be included. This concludes two thirds of Katzkes approach and is sufficient for a black-box evaluation.

For a more rigid evaluation, design documentation, test protocols and source code can be submitted. The information from the developer provides the basis of the evaluation. The security policy is founded on either a statement by the developer or as a result of the security objectives.

Evaluating COTS demands a more standardised approach. Neither the developer, nor the evaluator knows the exact environment in which the object is going to operate or how it will be used. To provide continuity and for being able to compare the result with other similar COTS, a standardised environment can be used. The environment should be based on general demands that can be assigned to the purpose of the object and restrictions that come along with it.

The other half of the knowledge part is up to the evaluation facility. They should be able to translate known vulnerabilities presented in new approaches, i.e. being able to dissect the security enforcing functions provided to counter known threats and remain up-to-date with new vulnerabilities.

4.2 Methods

Since there is no predefined method that suffices, the evaluator need to decide which approach is suitable to the needs of what to be accomplish depending on when, during the development process, the measure is taken place. Commonly, during development it is the developer who evaluates the product. Third-part evaluation is performed when the product is completed, though different stages of the development may be taken under consideration.

For an evaluator, the specific task that is going to be measured decides the method. If you want to find out if a firewall is resistant to penetration, penetration tests can be performed. If you want to confirm your security policy, logic and flowcharts may do the job.

The methods can be practical or theoretical. Practical methods regard testing the product with tools, whereas theoretical methods are a bit more vague. They are based on general assumptions and experience, followed up by testing.

4.2.1 Tests

Penetration is one form of testing that can be performed on a fully developed firewall. One way of performing penetration tests is by using commercial tools, for instance scanning tools. A scanning tool checks for open ports that can be used to send packets to the system. There are several tools available, such as ISS or SATAN. Other tools that often come with the operating system is used, for instance variations of the ICMP protocol such as *ping* and *traceroute*. These tools where originally designed for a network administrator to maintain the network, but have proven useful for an attacker as well.

The simplest attack to perform is a DoS attack. The result of a DoS attack is strictly destructive, even though the attacker may have greater plans other then just causing a breakdown. A successful DoS attack causes the firewall to shut down or reboot, which in turn can be expensive if the owner is running electronic commerce. The interesting part, in an evaluation point of view, is how the firewall reacts when closing or rebooting. Does it still enforce the security policy as intended or does it open up completely?

Performing a penetration test in its most simple form constitutes of making an invalid packet getting through the firewall. The purpose is to identify a machine on the other side of the firewall, thus being able to recreate the topology of the network. This is not a direct damage to a system, but it is the first step in a larger attack.

To escalate the test a red team is organised. It is a group of people that will try to penetrate the firewall and to achieve a given goal within the system. For instance, they could try to take control over a computer or compromise a server.

Red team penetration a firewall is a more complicated act. It usually demands that the attacker have some knowledge of the firewall. A poorly configured firewall may provide information to the attacker when he tries to access the system behind it. Knowing what type of firewall is being used, the attacker can study public reports on vulnerabilities or receive guidance from other hackers who have penetrated the same kind of firewall before.

By using common vulnerabilities, the evaluator should confirm whether or not the developer have implemented proper countermeasures to avoid them. Also, a

DoS attack will answer the question of how the firewall reacts when forced to reboot.

4.2.2 Assessment

When a developer designs a firewall, he decides what it is supposed to do well. As always with a product that can do several things, there are some properties that are prioritised. Speed is always a crucial factor. A slow firewall will probably not be introduced on the market even if it is regarded as very secure. In some ways, a firewall can be an obstacle in the day-to-day use of the network. The developer has to make a trade-off between capacity and security.

A major problem with computer security is the inherited problems [CCZ00, 2]. Inheritance is due to shared protocols, implementations or components. Products using the same protocol or basic design idea shares the same fundamental problems that comes from that protocol, regardless of what name is printed on the component. The same goes for implementations. Some developers use Microsoft NT or Unix operating system in their firewall. As seen in the example further down even software components may suffer from inherited problems. A firewall can comprise of several components, both hardware and software. Hardware components may be bought from other developers or be of own design. The origin of a bought component may be unclear. It is not uncommon within the computer industry (or any other industry for that matter) that components are sold using different names. There are commercial benefits of having components with the selling company's name on it. Due to this, the strength and weaknesses of that component may be apparent in several firewalls, even though they seem to have different origin.

By introducing a newly developed system many of these inherited vulnerabilities can be taken care of at an early state. Unfortunately, the leverage gained may only be temporary. There are benefits of using established products or implementations. They may not be perfect, but they have been scrutinised by independent parts for a period of time, hence their vulnerabilities are known and can be dealt with. Introducing a new implementation is a bit of a wild card.

An example should provide some clarification on this.

> There are several implementations of the TCP/IP network standard. Both Unix and Windows NT use it. It is a low-level network protocol, which is difficult to upgrade. Due to this, the original implementation in an operating system usually remains the same over time. Unix has the longest history of TCP/IP use and therefore has the longest experience of the problems surrounding their implementation and how to handle them. When Microsoft designed their NT system they reimplemented the whole TCP/IP stack from scratch, resulting in several errors that Unix discovered and adjusted long time ago. This is not an impossible problem to solve. A developer has to define the errors and produce evidence of how they are encountered [CCZ00, 3].

The task of the evaluator is to confirm that the security enforcing functions installed are sufficient to counter known vulnerabilities and does not result in new ones. It is in the developers' best interest to produce documentation that describes the security enforcing functions and how they avoid or hinder these vulnerabilities to be exploited. If the system is newly developed, the evaluator has to be able to confirm that "old" problems are not inherited and new ones are not introduced.

4.3 Results

What kind of result can we expect, when performing measurements? The objective part, the actual method of measurement, gives us the unit of our measure. By using metrics, we can transform that unit into something meaningful – from a security point of view.

What is the result of the evaluation? Was the penetration test successful? If it was, can the product obtain any assurance approval? If it was not successful, does it mean that it is impossible to penetrate, or that the team was not good enough?

Questions like this will always arise when evaluating security. It is a constant race between developers of security products and the attackers. If security is binary (secure or not secure) the developer has to provide documentation of evidence that every possible and hypothetical vulnerability is addressed. Then again, new vulnerabilities are discovered regularly, which means that a certified firewall may be deemed insecure in a matter of months or even weeks [PFL01].

When using scales, we circle the binary properties. The scale can be adapted to fit the purpose of the metric, which makes for a more dynamic evaluation and a result that is easier to present. The scale can, for instance, be set to represent the amount of security enforcing functions implemented or the time it takes to penetrate a system. The choice is up to the evaluating party.

In whatever way the result is presented, it has to follow basic scientific rules. These include, apart from validity and reliability, that the result is repeatable. The result should be the same if the same firewall is put through the same evaluation process again.

4.3.1 Validity of the result

The main issue is not whether a scale or threshold is being used. The important issue is what constitutes the basis for setting the scale or threshold. I see three possible angles that may decide whether to use a scale or threshold:

- Purpose or policy,
- Environment, or
- The security enforcing functions.

Is it the purpose of the product that forms the evaluation criteria? From a firewall point of view, this means that the security policy has to be able to be enforced by the firewall. In order to measure the capability to do so, the security policy either has to be known or the firewall has to be able to enforce any given security policy. The result of such an evaluation can only be binary – either the firewall can handle any security policy or it cannot.

The operating environment can be used as reference. This allows for a scale to be used, since different environments can be referred to as having different needs of security. For instance, the threats and needs are different in government and commercial organisations. The government, e.g. the intelligence community, need to keep information secret for a long period of time or may want to access classified databases from a remote location. Commercial organisations need to provide secure monetary transactions, e.g. by the use of credit card numbers. As a private user of a computer attached to the Internet, the information may not be as sensitive as in the cases mentioned above. Still, the machine is meant for private use, hence it should be protected from hi-jacking. Even if the computer does not contain any useful information for an attacker, the machine itself can be used as a base to attack others.

Using the environment as a reference, a scale may very well be suitable to describe the level of security offered. This is useful when comparing COTS. Since the developer does not know the specific operational environment, an environmental-scale can suffice as a yardstick. The product will be tested to see if it is suitable for a certain environment. Again, the security policy has to be set in some standardised manner in order to be able to compare different products.

Finally, if the product itself should stand as the reference, the security enforcing functions are of primary concern. The functions themselves may not be hard to evaluate, but to form a result that can be compared with other similar products may be tricky. In this case, the environment is of no concern, nor the policy, even if a standardised security policy is being used during evaluation. There will be a lot of demands on the operational environment for the evaluation result to hold. The result can be presented either as binary or through a scale, though it is the claims of the developer that really determines the level of security. Using the security enforcing functions as the result provide a reference that may well be understood by system administrators but it is harder to use when presenting the result for non-technical staff.

Whatever method is being used or what properties that sets the scale or threshold, security is still an act with a short expiration date. A policy has to be re-examined regularly and the logs of the firewall have to be followed-up. The day-to-day security work is often about finding out afterwards what has happened and correcting the failures.

4.3.2 Reproducibility of the result

It is difficult for any kind of scientific method to be reproducible when it is based on experience. The knowledge of the used protocols and their implementations grows as long as they are around.

Penetration tests are especially hard to reproduce in a fair manner. Every attempt to penetrate a system is a lesson learned. The knowledge of the technology in general and the specific system at hand is either confirmed or enhanced. When a security breach is discovered, the tools and methods are adjusted, hence becomes more effective than before. There is no saying that the same firewall will pass the test twice if the developer has not kept up with the discoveries.

4.3.3 Control of the result

Computer science is a fairly new science, which unfortunately result in relatively low experience. There are no laws similar to the laws of nature to prove an evaluation correct. There is still a lot to learn and progress is being made rapidly – both by developers and hackers.

Control is achieved by clearly stating how the evaluation is performed and what methods are being used.

4.3.4 Terminology

To avoid misunderstandings and unnecessary debates, a terminology should be produced as a guide to how to express the abilities of the firewall. The same terminology should be used when delivering the result of the evaluation.

4.3.5 Ethics

During the evaluation process, the evaluators study material not meant for public use, for instance design documentation or source code. These are valuable documents for the evaluation, but the content should not be reviled. It is important that the result of the evaluation is presented in such manner that the result is clear, but does not expose business critical information.

The staff of the evaluation organisation should be bound by some code of ethic, prohibiting them from, for instance sharing their experience in a newsgroup.

4.4 Criticism of the model

The model presented is very general. It is aimed as a reference tool for comparing other evaluation schemes. Still, at such a general level some of the difficulties of security evaluations are obvious.

As seen, the value of penetration tests are questionable. An unsuccessful penetration test, in the sense that the firewall could not be penetrated, is still no evidence that a vulnerability does not exist. Reproducibility, which is a receipt that the evaluation method is correct, is especially difficult to confirm due to the technology itself. The technology evaluated is unstable and progressing rapidly.

Since the reference model is an abstraction of security evaluation schemes it surface above these problems, thus being able to evaluate them in an objective way.

5 Discussion of methods based on the framework

With help of the general model presented in the previous chapter, we shall now see how the security evaluation models from chapter three comply with it.

5.1 Knowledge

One of the main differences between government and commercial security evaluation is the information supplied by the developer. The reason is the level of the intended result.

5.1.1 The object

There is a vast difference between the documentation provided to government evaluation compared to commercial evaluation. ICSA FWPD and Firewall Checkmark Criteria conduct black-box evaluation, which does not take the design and implementation under consideration. They only need the documentation delivered with the firewall when sold.

The amount of documentation provided to ITSEC and Common Criteria depends on the intended assurance level. A higher assurance level naturally demands deeper knowledge of the TOE. Design documentation is required at an early state though the formality increases with the assurance level. Source code and object code is required for the top assurance levels.

5.1.2 The environment

In order to provide consistency, the commercial models provide a predefined environment with a set security policy. They can allow for this since their evaluation is specially designed for firewalls. On the other hand, this transfers some of the evaluation problems to a customer. He has to determine whether their environment and policy is applicable for his purposes.

Again, government methods are more dynamic. The environment and the security policy are stated in the security target. One might argue that evaluators of government schemes does not know more about the operational environment than their commercial counterparts, which in some ways are true. The edge provided to the ITSEC and Common Criteria evaluators is the product rationale of the security target. Here information about technical configuration and supporting software can be included and taken under consideration. A developer may very well provide this information together with a commercially evaluated firewall, but is not taken under consideration during evaluation.

5.1.3 The evaluators

The competence of the evaluators is hard to estimate. One way of getting a hint is by seeing how well the market has adapted to them, e.g. how many developers have had their products evaluated by them.

5.2 Methods

5.2.1 Assessment

The thorough analysis of the design and, for higher levels, the source code of the TOE provide information to the evaluators that may not be obvious or, for that matter, discovered during the test phase. Studying the documentation from the development and testing done by the developer may find errors that eventually would be discovered anyway, though probably not by someone evaluating the firewall.

One important task for the evaluators is to determine the consistency in the design of the firewall. Gaps in the design may lead to weaknesses and in the end an exploitable vulnerability.

It is the rigor in the analysis that gives ITSEC and Common Criteria the ability to discover problems that may not have been found with the tools used by ICSA FWPD or the Firewall Checkmark Criteria. They are limited to the efficiency of their tools. This is not to say that the tools are bad, it is just that they can only find what is already known. ITSEC and Common Criteria can, to some extent, foresee possible problems.

5.2.2 Tests

ICSA FWPD and the Firewall Checkmark Criteria issue a black-box test on the firewall. They use commercially available tools as well as tools developed by them selves. The result is a consistent approach to all firewalls, even though downside of only using penetration testing is obvious from previous chapters.

ITSEC and Common Criteria uses the same types of tools, but have the ability to specify what is going to be tested individually depending on the outcome of the development and test analysis.

5.3 Result

The ambition of the evaluation schemes is reflected in the manner they present their result. ICSA FWPD and Firewall Checkmark Criteria both have the ambition to produce a fast result, claiming a basic level of assurance. They have a specific goal to reach, hence their result is binary.

ITSEC and Common Criterias methods are more complex and therefore they let the developer decide how much effort that should be put into an evaluation. The result is presented as a scale showing the effort made to provide evidence of their claims.

ICSA FWPD is the only organisation that mandatory regards security evaluations as a process. They force the developers to keep up-to-date by conducting random re-evaluations, thus allowing the certification to remain through updates.

5.3.1 Validity of the result

Both ICSA FWPD and Firewall Checkmark Criteria use a predefined security policy. Since their method of evaluation consists of black-box penetration test, it is necessary in order to achieve a comparable result. On the other hand, this leaves no freedom to the developer. The result is based on the firewall capability to enforce their security policy in their environment. It is up to the customer to determine weather or not his environment is similar and if the certification is valid there.

ITSEC and Common Criteria base their evaluation on the described environment in the security target. This should result in a more accurate evaluation of the firewalls capabilities. Still, the customer has to decide if it applies to his needs, but the information supplied for evaluation should yield a more accurate situation.

Often developers of COTS use a protection profile as a reference when writing a security target. This is to achieve a compatible result that can be used when comparing different firewalls. Since the protection profile already has been evaluated, it functions as a guide for the developers, and provides some extra reference material for the evaluators. In turn, this should lead to a thoroughly developed product.

5.3.2 Reproducibility of the result

Since new security breaches are discovered on regular basis, it would be unfair to compare two firewalls evaluated, say, six months apart. New breaches demand new or modified security enforcing functions.

A penetration test is even harder to reproduce in a fair manner. Every attempt to penetrate a system is a lesson learned. The knowledge of the technology in general and the specific system at hand is either confirmed or enhanced. When a security breach is discovered, the tools and methods are adjusted, hence becomes more effective than before. There is no saying that the same firewall will pass the test twice if the developer has not kept up with the new discoveries.

Reproducibility is difficult for all the evaluation schemes, especially when it comes to penetration tests.

Common Criteria evaluators are obligated to produce documentation of how and what have been tested, in order to enable reproducibility.

5.3.3 Control of the result

Control is achieved by stating the conditions of the operational environment. The commercial methods apply their evaluation in a fixed environment, whereas the government run method uses the information provided by the security target.

ICSA FWPD takes it one step further by making the evaluation to a process over time. This is not way of enforcing control, rather a way to adapt to what cannot be controlled.

5.3.4 Terminology

Common Criterias class structure defines a terminology of how to describe the TOE and the actions taken by the developer and evaluator. No equivalence is found in the other evaluation schemes, which make Common criteria evaluations easier to follow and understand.

However, this does not constitute as a problem for ICSA FWPD or the Firewall Checkmark Criteria since they conduct black box testing.

5.3.5 Ethics

Security is about trust, which make the integrity of the evaluating organisation crucial. In many ways this should not constitute as a problem, but it is important that the result of the evaluation is published in a way not revealing sensitive information about the firewall, yet revealing enough information about how the evaluation is performed so a customer can decide if it suffice.

5.4 Summary of discussion

Table 5.1 is a summary of the evaluation schemes matched with the model presented in chapter 4.

The dynamic characteristics of government evaluation schemes leave the developers more freedom to produce a firewall for a specific environment. They also demand a more thorough description of the firewall and its features in order to assure an accurate evaluation. In turn, this leads to a higher level of control during the evaluation.

	ITSEC	**CC**	**ICSA**	**FCC**
Knowledge				
Object	*Dynamic*	*Dynamic*	*Basic*	*Basic*
Environment	*Varies*	*Varies*	*Predefined*	*Predefined*
Method				
Assessment	*Yes*	*Yes*	*No*	*No*
Tests	*Yes*	*Yes*	*Yes*	*Yes*
Result				
Validity	*Dynamic*	*Dynamic*	*Basic*	*Basic*
Reproducibility	*Possible*	*Possible*	*Possible*	*Possible*
Control	*Yes*	*Yes*	*Yes*	*Yes*
Terminology	*No*	*Yes*	*No* *(not needed)*	*No* *(not needed)*
Ethics	*Good*	*Good*	*Good*	*Good*

Table 5.1. *Result of evaluation according to the reference model.*

As said before, ethics is hard to judge. All methods publish the result along with a short description of the firewall – much like a description the developer would print on a product information leaflet. Looking at their acceptance, one can see that Common Criteria is gaining ground. More products are submitted to evaluation with the intention of receiving a higher certification. The level needed today to keep up with competitors is EAL 4. It is not unusual that the same firewall is submitted to different evaluation schemes. The most common firewalls on the market, for instance FW-1, are certified by Common Criteria as well as by ICSA FWPD.

6 Conclusions

The main problem for security evaluation is the lack of knowledge of the technique. It is not necessarily the hardware that causes the problems, but how the software parts work together with the necessary protocols. Vulnerabilities that may cause security breaches are discovered regularly, though often they are due to exploration of existing implementations and standards, rather than newly developed techniques of attacking.

Providing a model for security evaluation that preserves validity, reproducibility, control, terminology and ethics has proven difficult. The development speed of the computer industry makes it hard to evaluate the consequences of newly developed services and how they affect the security functions of a firewall. As a consequence, the security of a firewall cannot be guaranteed. This makes control, and thus reproducibility, the hardest property to achieve since the situation rapidly changes.

Control is achieved by stating the security policy and defining the tools and method used during evaluation. Black-box approaches, as with ICSA FWPD and Firewall Checkmark Criteria, puts higher demands on standardisation of the inputs of the test. They achieve control by using a predefined security policy and, in ICSA FWPDs case, a description of at least some of the tools used. To be able to reproduce the evaluation, the same versions of the tools as where used the first time must be reused.

Common Criteria and ITSEC allows for the developers to set the security policy in the security target, thus letting the developers establish what is to be the controlled environment. As long as the security target remains the same, and no new vulnerabilities have been found in the mechanisms, there should not be any problems in reproducing the same result.

Government evaluation methods are superior to commercial evaluation. They provide a much more rigor evaluation by examining the development of the firewall. This makes it possible for them to discover covert channels or gaps between the design and implementation. From EAL 5 of the Common Criteria, the rigor of the evaluation is good. At these levels the source code is, at least partially, submitted and covert channels are analysed, giving the evaluators the possibility to predict future vulnerabilities. This should suffice for most common organisations not dealing with classified material.

The black-box approach taken by ICSA FWPD and Firewall Checkmark Criteria is restricted to the obvious errors found by the tools used. They have no means of confirming the correctness of the firewall, nor the possible shortcomings of the tools. Their primary concern is to provide a fast evaluation, with the purpose of showing that the firewall should withstand the most common attacks.

An issue for all security evaluation schemes to improve is the procedures of revalidation. Today ICSA FWPD is the only scheme that allows the product to maintain its certification through updates. It is understandable that the other schemes do not want to do so since security breaches often occurs when modules interact. Allowing for maintained certification is in many ways a bold statement even though random checks are performed two to four times a year.

The owner of a network should be able to feel assured when using a firewall certified to EAL 5/E4, provided that it is configured according to the directives in the security target and, not to say the least, the users of the network complies with the security policy.

References

[ALG01] Alger, J I: *On Assurance, Measures, and Metrics: Definitions an Approaches*, 2001
www.acsac.org/measurement/position-papers/index.html,
visited 29 March, 2001

[BAR01] Bartol, N: *IA Metrics Development and Implementation*, 2001,
www.acsac.org/measurement/position-papers/index.html,
visited 9 April, 2001

[CC] Common Criteria
www.radium.ncsc.mil/tpep/library/ccitse/ccitse.html
visited 13 June, 2001

[CC, 1] Common Criteria, version 2.1, part I, p 39,
www.radium.ncsc.mil/tpep/library/ccitse/ccitse.html
visited 18 June, 2001

[CCZ00] Chapman D B, Cooper S, Zwicky E D: *Building Internet Firewalls (2^{nd} edition)*, O'Reilly & Associates, Inc, Sebastopol, 2000, p 102, ISBN: 1 - 56592 - 871 - 7.

[CCZ00, 1] Chapman D B, Cooper S, Zwicky E D: *Building Internet Firewalls (2^{nd} edition)*, O'Reilly & Associates, Inc, Sebastopol, 2000, p 4, ISBN: 1 - 56592 - 871 - 7.

[CCZ00, 2] Chapman D B, Cooper S, Zwicky E D: *Building Internet Firewalls (2^{nd} edition)*, O'Reilly & Associates, Inc, Sebastopol, 2000, p 69, ISBN: 1 - 56592 - 871 - 7.

[CCZ00, 3] Chapman D B, Cooper S, Zwicky E D: *Building Internet Firewalls (2^{nd} edition)*, O'Reilly & Associates, Inc, Sebastopol, 2000, p 31, ISBN: 1 - 56592 - 871 - 7.

[DOD85] TCSEC, 1985
www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html,
visited 7 March, 2001

[FCC] Firewall Checkmark Criteria
www.westcoast.com/checkmark/cm_fwc.html
visited 22 March, 2001

[GIB01] Gibson, S: *The Strange Tale of the Denial-of-service Attacks Against GRC.COM*,
grc.com/dos/grcdos.htm,
visited 18 June, 2001

[HAHU99] Computers & Security, vol.18, No. 2, pp 165-177, 1999,
ISSN 0167-4048

[ISO17799] The ISO 17799 Service & Software Directory
www.iso17799software.com/,
visited 27 April, 2001

[ISSRR] Workshop on Information-Security-System Rating and Ranking,
May 21-23, 2001, Williamsburg, Virginia,
www.acsac.org/measurement/index.html,
visited 8 June, 2001

[ITSEC] Information Technology Security Evaluation Criteria
www.itsec.gov.uk/,
visited 26 April, 2001

[ITSEC, 1] ITSEC Developers' guide – part III: *Advice to Developers*, p 27.
www.itsec.gov.uk/docs/formal.htm,
visited 26 April, 2001

[ITSEC, 2] ITSEC Developers' guide – part II: *Referince For Developers*, p 142.
www.itsec.gov.uk/docs/formal.htm,
visited 9 March, 2001

[KAT01] Katzke, S (Dr.), *Security metrics*, 2001
www.acsac.org/measurement/position-papers/index.html,
visited 29 March, 2001

[MAX01] Maxion, R A, *Dependable measurement*, 2001,
www.acsac.org/measurement/position-papers/index.html,
visited 11 April, 2001

[NCSC99] Common Criteria Certified protection profiles
www.radium.ncsc.mil/tpep/library/protection_profiles/index.html,
visited March 20, 2001

[PFL01] Pfleeger, C P, *Newsletter*, 2001
www.counterpane.com/crypto-gram-0003.html,
visited 24 April, 2001

[PFL97, 1] Pfleeger, C P, *Security in computing*, Prentice Hall, 1997, p 5,
ISBN: 0-13-185794-0

[PFL97, 2] Pfleeger, C P, *Security in computing*, Prentice Hall, 1997, p 309 - 312, ISBN: 0-13-185794-0

[PFL97, 3] Pfleeger, C P, *Security in computing*, Prentice Hall, 1997, p 314,
ISBN: 0-13-185794-0

[STO01] Stoneburner G, *High assurance ≠ More secure*, 2001, www.acsac.org/measurement/position-papers/index.html, visited 29 March, 2001

[TCSEC] Trusted Computer System Evaluation Criteria www.radium.ncsc.mil/tpep/library/tcsec/index.html visited 13, June 2001

[TRUE] ICSA labs www.icsalabs.com/html/communities/firewalls/certification/criteria_3.0a.shtml visited 6 June, 2001

APENDIX A

What is a firewall

There are people arguing on what is a firewall and what is not. In this report I shall comply with the definition made by Chapman, Cooper and Zwicky [CCZ00], that a firewall is a component or a set of components that restricts access between a protected network and the Internet, or between other sets of networks.

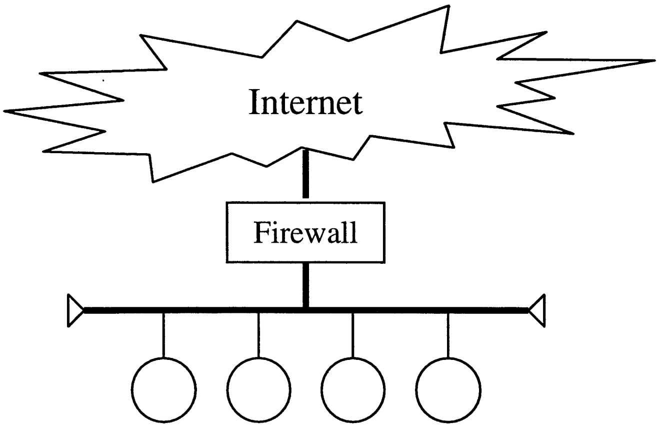

Figure A.1. *A firewall between a private network and the Internet.*

For simple editorial reasons, I will refer to a firewall as being situated between a private network and the Internet, even though I am aware of other combinations.

The purpose of a firewall is to control the traffic to and from a network. Usually when describing a firewall one is often concerned about how it should protect the network from outside dangers. A firewall can of course do this but is can also, to some extent, protect the network from its users. The same techniques used to control inbound traffic can be used to disallow outbound traffic. This can be good if, for instance a Trojan horse has entered the network and start sending out information. The firewall can detect that a new program not is trying to access the Internet, hence ask for confirmation that this is correct.

The actual technical solution of the firewall varies. There are two main categories of building techniques:

- Screening routers, and
- Proxy gateways.

A screening router is the simpler, cheaper and fastest of the two. Basically, it is a machine with two network interfaces – one facing the private network and the other facing the Internet. The purpose of the router is to determine whether or not to forward packets from one interface to the other, based on a specific set of rules.

As the screening router operates in the network layer, the security functionality in a router is limited to inspecting the header of each packet. It can, for instance allow/disallow a packet based on source/destination address, protocol type, source/destination port or the setting of the flags in the header.

There are modified screening routers, called stateful packet inspection firewall, allows for gathering packets, thus learning more about the content of the transmission, before passing them on to the other network. This increases the security level, by stopping or at least reducing fragments on the private network.

When using a proxy gateway, all traffic from the inside to the outside and vice versa travels through a proxy. A packet is sent to the proxy, which decides (depending on the implemented set of rules) whether or not to forward the packet. Proxies operate at the application layer, which allows for a more detailed inspection of the packets. Due to this, a proxy can determine the content of the packet, for instance if it is an email containing plain text or an MS Word document. The cost of more explicit rules is speed.

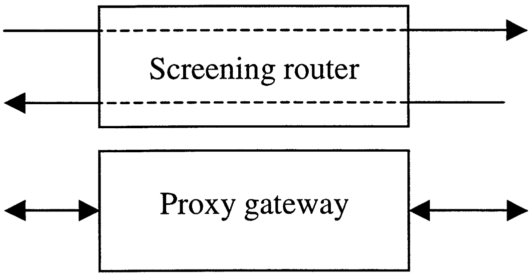

Figure A.2. *Screening router and proxy gateway.*

There are numerous combinations of these techniques. Proxies are the dominating technique since it is more flexible and allows for a more specific set of rules.

Usually corporate networks consist of three different networks: the Internet, a DMZ and the private network. The demilitarised zone, DMZ (actually an acronym from the Korean war), is an intermediate network between the Internet and the private network. Here resides, for instance, web servers, mail servers and DNS servers – systems that cannot be fully trusted, yet needs as much protection as possible.

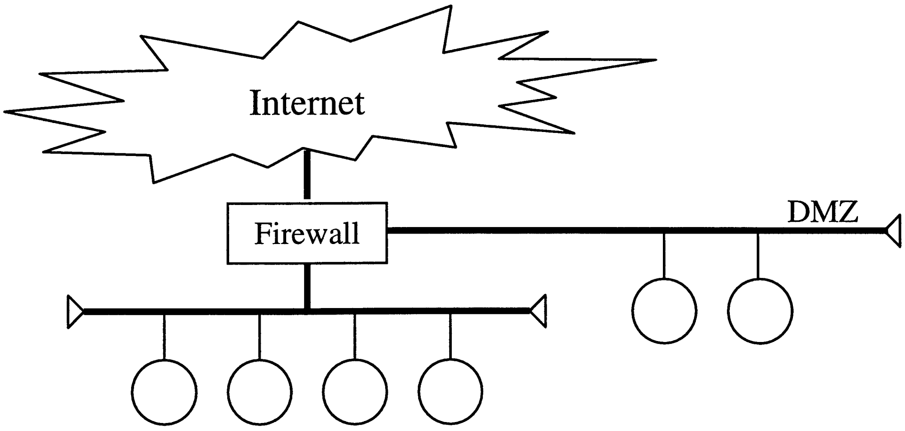

Figure A.3. *A larger firewall system with different protective areas.*

APPENDIX B

A brief presentation of the TCP, UDP and ICMP protocols

Computer communication is usually described using the OSI reference model or the more common TCP/IP-suite. This appendix will briefly present three protocols residing in the transport layer of the TCP/IP-suite, and how they can be used to attack a firewall.

The Transfer Control Protocol (TCP) is a connection-oriented protocol, which means that it guarantees delivery. It connects to a remote host using a three-way handshake similar to placing a phone call. The receiver has to acknowledge the call before transmission can begin.

It is during the handshake procedure the protocol is vulnerable for hi-jacking. This is a difficult attack, though not impossible.

The Universal Datagram Protocol (UDP) is connectionless, in many ways similar to sending a postcard. You do not know if or when it will arrive. It is easier to disguise the sender since no formal connection has been established, making it a popular protocol to use for attacks.

When a packet fails to reach its destination, a control packet is returned to the sender using the Internet Control Message Protocol (ICMP). It is a service protocol with many tasks. Several network tools such as *ping* and *traceroute* uses ICMP.

A properly configured firewall should be very restrictive about allowing the ICMP to return with information from, or worse behind, the firewall. The information can be used to identify machines behind the firewall.