Zero effort security for the home PC users?

Terje Risa

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2008

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

In today's technological society, where computers are increasingly used to access all kinds of information over the Internet, it is important to protect the computer from the hazards of the Internet. Whether it being Internet related crime, such as online banking accounts being stolen or malicious software compromising a system, it is vital that computer users protects their computers. For the common home computer users, this can be a very difficult subject requiring much effort. Home PC users needs security solutions which minimize their effort needed, and at the same time provides effective and usable security.

With the vast variety of security products available for computer users, it is important that theses products are secure and usable. For evaluating if the different products are usable and secure, there is a need for methods which can highlight this. This thesis will therefore investigate some of today's security products available for the home computer user, to see if the method used in this thesis is suited for evaluating the security and usability of the products.

The results of the work carried out in this thesis are a methodology for evaluating the security and usability of some security products. This method will be tested through some experiments and will be based on the work and knowledge of others. From this methodology some of today's comprehensive security product's usability and security effectiveness will be visualized.

Sammendrag

I dagens teknologiske samfunn, hvor datamaskiner i voksende grad blir brukt til å aksessere all slags informasjon over Internett, er det viktig å beskytte disse maskinene fra farene på Internett. Enten det er Internett relatert kriminalitet, sånn som nettbank tyveri eller ondskapsfull programvare som kompromitterer et system, er det viktig at datamaskin brukere beskytter sine maskiner. For den gjennomsnittlige hjemme PC bruker, kan dette være et veldig vanskelig tema som krever mye anstrengelser. Hjemme PC brukere trenger sikkerhetsløsninger som minimaliserer kravet til innsats, og på samme tidspunkt sørger for både effektiv sikkerhet og brukervennlighet.

Med det store mangfold av forskjellige sikkerhetsprodukter tilgjengelig for datamaskin brukere, er det viktig at disse produktene er både sikre og nyttige. For å evaluere om de forskjellige produktene er brukervennlige og sikre, trenger man metoder som kan fremheve dette. Denne masteroppgaven vil derfor undersøke noen av dagens sikkerhetsprodukter som er tilgjengelige for hjemme PC brukere, for å se om metoden denne oppgaven bruker er passende for å evaluere sikkerheten og brukervennligheten av produktene.

Arbeidet gjort i denne masteroppgaven har ført frem til en metode for å evaluere sikkerheten og brukervennligheten til noen sikkerhetsprodukter. Denne metoden vil bli testet gjennom noen eksperimenter og vil være basert på andre folks arbeid og erfaringer. Ved bruk av denne metoden vil noen av dagens omfattende sikkerhetsprodukters brukervennlighet og sikkerhets effektivitet bli visualisert.

Acknowledgements

This Master's Thesis will complete my degree of Master of Science in Information Security at Gjøvik University College (GUC) and finalize my 5-years as a student at GUC.

I would like to give a big thanks to my supervisor Professor Einar Snekkenes for great guidance throughout the work on this thesis. I would also want to thank the other people that has helped me through the work with this thesis; Frode Volden for guidance regarding usability, a special thanks to all the participants for taking time to perform the experiment, all my classmates at GUC, the library staff for their help, Andreas Clementi for permission to use parts of the results from AV-Comparatives and all those who I have not mentioned here.

- Terje Risa, 25th June 2008

Contents

List of Figures

List of Tables

1 Introduction

With the increasing use of Internet to access sensitive information, online banking and electronic commerce, is the need for proper protection of home computers a pressing issue. Home computers are becoming a more valuable and easier target for malicious users than corporate computers, and thus increasing the threat against home PCs. The service providers, like the authorities responsible for the information portal 'Minside'[1], ensure adequate protection for their services, but not for the computer accessing these services. Some service providers gives out security software, like anti-virus programs to their customers, but very often is the user left alone to properly protect their home computer.

Since many services available through the Internet are commonly accessed with a home computer, and these services make the home PC a more valuable target for malicious attackers, should the service providers help users protect their computer. Service providers are often interested in making their service adequately secure without this affecting the usability, and one way of strengthen the total security of a service is by helping the end-users protect their PC.

1.1 Topic covered by this thesis

The topic of this thesis is home computer security and to investigate if some chosen security suites can help toward zero effort security for the home PC users. It is important that security products available for home computer users are secure enough to protect against all kinds of relevant threats and at the same time be usable for the common user. This thesis will investigate the two often conflicting dimensions of usability and security in some products. There will be deployed resource economical methods for evaluating the usability and security effectiveness, to see how appropriate the method is for evaluation.

1.2 Keywords

Computer security, home computers, security products, security effectiveness, usability, product evaluation, vulnerability, threat.

1.3 Problem description

Security in online banking systems and information portals containing sensitive user information has been a very important subject. This has resulted in more secure solutions for the users, for instance the use of one-time password in conjunction with online banking. This focus on improving the security of corporations and businesses, together with many other factors, has lead to security threat shifting toward including attacks against home computer as well. Since corporate computers have become more difficult to attack, together with the propagation of home computer connecting to the Internet with broadband connection[1], has home computers

[1] 'Minside', 'MyPage' in English, is a governmental web portal that brings public service offerings together and represents an opportunity for a simple and effective dialogue with the public sector.

become a more valuable target for malicious attacks. Bruce Schneier, a famous and reputed computer security specialist, reports in [2] that another 2004 trend will be expected to continue in the years to follow, namely computer and Internet related crime. With the increase in computer and Internet related crime, will home PC users be a natural target that needs to be protected.

Unfortunately is often the home PC security neglected when for instance securing an information portal or online bank, even though many users use these computers when accessing the sensitive information. Some online banking companies and other service providers provide their users with anti-virus software, but this does not ensure that the user is protected enough. To improve on the problem of home computers being contaminated with malware and becoming part of botnets, user's need user-friendly security products recommended by service providers. It is particularly important that the security products are usable for the common home computer user, for it to be used. So for the service providers to be able to recommend different solutions suited for the home computer users, is there a need for resource economical methods for evaluating how good these products are with emphasis on security and usability.

1.4 Justification, motivation and benefits

With the increasing use of sensitive information accessed via home computers, the service providers need to continually improve the security and defense of their product. One important step in the direction of getting satisfactory protection is not only to secure the service provided, but also help to secure the end-users. This would greatly increase the total security, and would help the users from getting their sensitive information leaked to potential attackers.

An article by Dourish et. al. [3] suggests that people often find, and trust, some external entity when managing practical everyday security problems. This could for instance be an organizations security team, a friend that is likely to have more security and technically expertise or information found in different tests. In [4], Cialdini describes several psychological principles that directs human behavior, the principle of *authority* suggests that listening to authorities for guidance is a common human behavior. With this in mind is there a real possibility that many people will start using security products that service providers have recommended, especially if the products have been identified to be usable and practically secure. In the master's thesis [5], Andreassen performs a questionnaire where 75% of the participants said they were interested in security measures and material made available through services like 'Minside'. This indicates that the general end-users are interested in security products recommended by their service provider.

> "Systems must be not only secure, but usably and practically secure."
> -Dourish et. al. [3]

1.5 Research questions

This thesis' setting is home computers, and the research questions that have been considered relevant for this thesis are:

1. To which extent is zero effort security for the home PC users possible with some of today's security products?
   - 1.1 What method is appropriate for evaluating the usability of these products?

1.2 What method is appropriate for evaluating the security effectiveness of these products?

1.3 What trade-off is there between security effectiveness and usability in today's security products?

2. Which vulnerabilities and threats are of current interest?

1.6 Planned contributions

With comprehensive all-in-one security products available for common computer users, is it important that these products are both properly secure and usable for them to be used by the general public. For being able to evaluate such products with regards to both security and usability, should there be standardized accepted methods for measuring these attributes. This thesis will try to assess the problem of evaluating security and usability in some of today's security products suited for the common home computer user. This will be done too look into if zero effort security for the home PC users is possible with such products.

The method of how to assess the effectiveness and usability of the security products is an important contribution, this can shed a light on the difficult subject of security and usability and how to evaluate it. This contribution might be more important than just the outcome of the evaluation and the results of the different products. Because that products changes, sometimes quite significantly, from different versions and newer releases, while the method of assessing the usability of the products does not differ so much.

This thesis will also hopefully help service providers realize the importance of helping their customers secure their computers, and possibly based on this thesis' product evaluation making it easier to recommend different security products fitted for the home computer users.

2 Overview of relevant security concepts

To understand the problems of home computer security, should one be aware of some fundamental issues of information and computer security. There will be a short description of some basic security terms and concepts. Then there will be a section on different security mechanisms commonly used for protecting computers, together with security packages available for the home computer marked. It is important to understand how the different security mechanisms work for them to be tested as correctly as possible.

2.1 Computer Security

Computer security can be defined as the art of protecting computers from danger, or described in other words, making sure that the desired status quo is not threatened or disrupted in any ways. Common ways of ensuring that a computer is protected from danger, is to implement known principles and strategies when protecting the computer. Defense in depth is one such practice of securing assets, where the security is implemented in layers in order to prevent, detect and respond to the danger presented[6].

A well known fact in computer security is that people often are the weakest link. The developers of security mechanisms seems to sometimes forget this fact and focuses on what could possibly go wrong instead of focusing on what probably will go wrong[7]. This leads to security failures because of human- and implementation errors, which should (and possibly could) be avoided if the security mechanisms was designed with focus on usability.

2.1.1 Why is computer security so hard?

To understand why computer security is such a difficult task, is there important to understand some fundamental facts. An attacker often only needs one security hole or bug to exploit, too get access to the victim's computer. If a person with malicious intentions finds a security hole in a computer which can be exploited, can this be used to install Backdoors and Trojans (see Section 3.4 for more information) to leave the computer compromised.

To protect a computer, the defender needs to plug every vulnerabilities, withstand scams, social engineering, new kinds of threats, etc. and continuously improve the security of the system. For security is not a one-time activity, but a continuous process[8]. All this must be done to prevent the attackers finding a way into the system. A paradigm that is commonly shown in everyday crime is that the easiest target is the one being attacked. This can very often be held true in computer crime as well, the easiest target is the one being attacked, if not other targets are of specific value for an attacker.

Because humans very often are the weakest link in computer security, is there a need for explaining some fundamental security issues and terms to help understand the problems of computer security.

2.2 Basic security terms

For understanding the concept of home computer security, it is important that some basic security terms are understood. The most important ones will be listed and briefly explained in this section.

Confidentiality, Integrity, Availability There exist three cornerstones in Information Security (IS), which almost every paper and article mentions and these three are listed below.

- Confidentiality - is the concealment of information or resources[9].
- Integrity - refers to the trustworthiness of data or resources[9].
- Availability - refers to the ability to use the information or resource desired by an authorized entity[9].

When listed together are they often abbreviated to CIA, and they are the fundamental characteristics of IS and computer security. These three cornerstones cover the most traditional areas of computer security, and their emphasis is on preventing unwelcome events[10].

If one accept the fact, that there is no such thing as a totally secure environment. That it is almost impossible to prevent all improper actions and intrusions to a system, should also some other aspects be considered when addressing computer security.

- Accountability -

  "Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party."
  -Orange Book (TCSEC)[11]

- Nonrepudiation - Provides unforgeable evidence that a specific action occurred[10].
- Reliability - Or dependability is the property of a computer system and the reliance users can justifiably place on the service it delivers[10].

Landwehr et al. [12] have developed a conceptual framework for dependable and secure computing, which encompasses some other attributes than mentioned above. These attributes are about the dependability of the system, and not so much on security.

- Safety - absence of catastrophic consequences on the user(s) and the environment[12].
- Maintainability - ability and ease to undergo modifications and repair[12].

Vulnerability, Threat, Risk and Exploit The concepts of vulnerability, threat and risk are key aspects in Information Security. They are defined in a variety of ways, where one version of them be presented here.

- Vulnerability - refers to a weakness in a system that could be accidentally or intentionally exploited to damage assets. This is often also referred to as a bug or security flaw and is essentially a mistake in specification, design or mostly mistakes in programming[10].
- Threat - is a potential violation of security[9].

- Risk - refers to whether an asset has a potential threat against itself and the likelihood of that threat being materialized[9] (i.e. if there exist a threat and vulnerability and the potential impact could be devastating, gives a very high risk). What countermeasures and how much effort and resources are used to combat risks are derived from a risk analysis. Often are mathematical equations, such as the function below 2.1 from [10], used together with frameworks like OCTAVE[13] to describe the risk level.

$$Risk = Assets \times Threats \times Vulnerabilities \tag{2.1}$$

- Exploit - is a program or a "cookbook" on how to take advantage of a specific bug, flaw or vulnerability to cause unintended behavior, like privilege escalation or denial of service (DOS) attack[6].

To visualize the security concept, have Common Criteria [14] made a general model (see Figure 1) that illustrates the concepts and relationships between owner, its assets and the threat and risk involved.

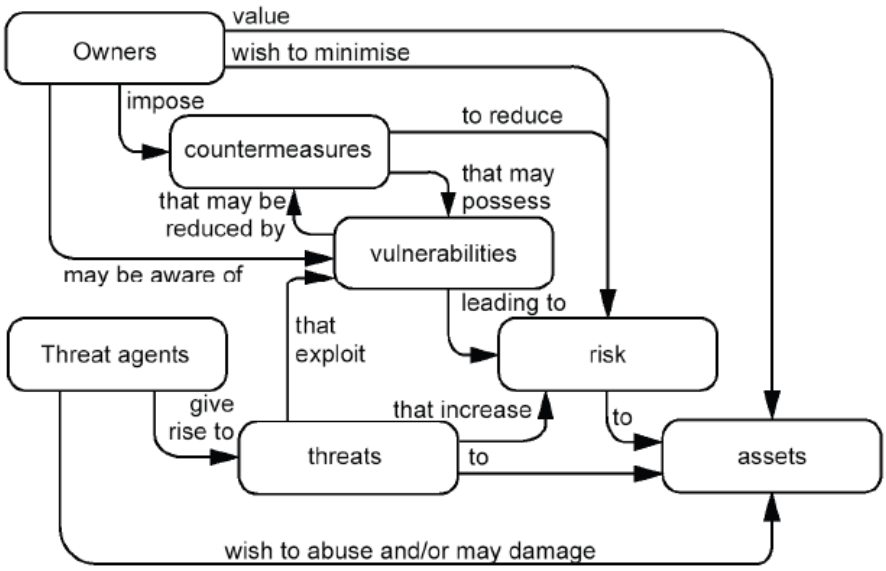

Figure 1: Security concepts and relationships, taken from [14].

2.3 Security mechanisms

There exist several security products tailor made for the home computer market. These products can vary from open source products to proprietary products, and utilizes some different security mechanisms. To understand what kind of protection the different solutions can provide, one needs to know how the different security mechanisms work.

2.3.1 Firewalls

A firewall is a device, software or specific hardware that is designed for limiting network access[15]. A firewall denies or permits packages based on a rule set. This means that every network package goes through the firewall, and each is checked up against the rule set. If a firewall has been configured correctly, could this provide very good protection, but if it is badly configured it will only give the users a false sense of protection. One common way of configuring a firewall is to set it too "'default-allow"', which allows all traffic through if it has not been specifically blocked. This way of configuring a firewall is very dangerous and should be avoided. The firewall should rather be configured with a "'default-deny"', which implies that all traffic through is blocked if not specifically allowed. This last way of configuring a firewall is much more secure, but unfortunately not as user-friendly since it either just drops packages (resulting often in tasks not functioning as they suppose from the users perspective) or ask the user what to do.

There exist several different types of firewall and firewall techniques, these different types and techniques works on different level in the network protocol stack.

Packet filtering Packet filters are often referred to as the first generation firewall, and they work by parsing the header of a network packet and determining on the source-, destination address or port number whether to drop or route the packet[15].

Dynamic packet filtering Dynamic packet filtering is when only the ports needed at a given time is opened. This means that the port is opened and connection is allowed through, but only for the duration of the connection. The firewall can also identify outgoing packet streams, and allow through incoming packets for the same connection[15].

Stateful packet filtering Packet filters that consider connection-state when deciding to pass or drop traffic, are called stateful packet filters. This enables the firewall to let through the traffic only if it is associated with an explicitly approved connection[15].

Stateful packet inspection Stateful packet inspection builds on the principles of stateful packet filtering, but it also includes analyzing the payload within a packet. For instance it can determine if the web traffic data is HTML or streamed music, and block streams of data according to the policy[15].

Application gateways Application-level filtering works on the application layer of the OSI model, and makes decisions based on the application data. This means that an application gateway, is an application-specific device which all inbound and outbound traffic must pass[16]. Some application gateway firewalls are application specific, meaning for instance that you have a firewall made for protecting e-mail only. In such a firewall is it special-purpose code that parses the mail, both the headers and the payload, and can possibly determine if the mail is of a malicious nature or not. Application gateway is a type of proxy server[1], and can validate an application specific request before sending it to the client or server[15].

[1] A proxy server is a server which forwards the clients requests to other servers, which allows clients to make indirect network connections to other network services.

Circuit gateways Circuit level gateways (sometimes called circuit relay) validates connections before any data is allowed transferred. This means that if a connection is determinate as valid will a session be opened and traffic will be permitted through only from the allowed source, and also possible only for a limited period of time. The validity of the connections can depend on source- and destination address, time of day, protocol or user and password. Circuit gateways contain a proxy mechanism, meaning that it possibly halts the original connection and originate a new connection from the firewall[15].

2.3.2 Anti-Virus

Anti-Virus software is a computer program that attempts to identify and neutralize malicious software, like viruses, Trojan horses and other types of malware (see Section 3.4 for more information). Anti-Virus programs utilizes different techniques, the most common ones are described below, for detecting as much different malware as possible.

Pattern-matching Most malware detectors use pattern-matching, or signature-matching, techniques to detect malware. Pattern-matching requires the anti-virus program to have some predefined information, a signature, about the specific virus, like for instance a unique string. This string or signature is used to define if the specific file in questioning is a virus or not. Pattern-matching are susceptible to obfuscation[17], which is a technique that creators of malware often utilizes when they want to create new and undetectable viruses.

Heuristic analyzer Heuristics classifiers are designed to detect new viruses, and do not need a signature. Instead is the behavior of a specific file in questioning inspected, and determined if it is of a malicious nature or not. Heuristics are not detecting viruses, it is simply looking for virus-like behavior. Some problems with heuristics are that it can produce quite a lot of false alarms, be time- and resource consuming and often still fail to detect new malicious executables[18].

Integrity checker Integrity checker scans the system and collects original “prints”, which are commonly stored as CRC-values, of files, boot sectors and system registry. These “prints” are stored in a database and compared against, to see if a potential virus have altered the CRC-values of file, boot sector or system registry[19].

2.3.3 Anti-Spyware

Spyware is some sort of software that is installed on a computer to intercept or spy on a user without the user’s knowledge about it. The term spyware is applied to any tracking technology, and Web Bugs[20] also can be classified as spyware. Web Bugs is often an invisible graphic, 1-by-1 pixel GIF, on a Web page or in an e-mail that is designed to monitor and track the user.

Adware, or advertising-supported software, is some form of software which displays or downloads advertising material to a computer. Adware can be used to provide legitimate software free of charge, but it also can be unwanted advertising and pop-ups from a user’s perspective. Some types of adware can be classified as a privacy-invasive software similarly to spyware.

There has been an enormous increase in anti-spyware programs available on the Internet. Some so called anti-spyware software have even been known to either be a sort of spyware,

or installing spyware/adware themselves. These rogue and suspect anti-spyware products often exploit users with scare tactics, deception and confusion, and a list of such known programs can be found on Eric Howes' website[2].

2.3.4 Intrusion Detection and Prevention Systems

To understand what Intrusion detection and prevention systems are you need to understand some key terms. First of, *a intrusion in computer science is a set of actions aimed at compromising the security goals of confidentiality, integrity or availability in a computing/networking resource. Intrusion detection is the process of identifying and responding to intrusion activities and intrusion prevention is the process of both detecting intrusions and managing responsive actions throughout the system it is protecting*[21]. Intrusion detection and prevention systems consists of sensors gathering data, a pre-processor that collects and formats the data, a detection algorithm that detects the different between "normal" and intrusive behavior and finally an alert filter that based on the behavior determines the response to the activity. In an Intrusion Detection System (IDS) will the alert filter based on the decision criteria estimate the severity of the intrusion and alert an operator. While in an Intrusion Prevention System (IPS) will the alert filter, based on the decision criteria respond to the intrusion, usually by blocking the activity for instance by dynamically adding a rule to the firewall[22].

There exists some different types of intrusion detection and prevention systems, the two main types are network-based and host-based. There are also application-based, which collects data from a specific running application and monitors the behavior of this application and target-based, which checks for alterations to a system or target. A network-based intrusion detection and prevention system (NIDS/NIPS) is as its name implies for protecting a network. This is often a dedicated piece of hardware that resides on the perimeter of a network, and thus protecting the internal hosts on the specific network. A network-based intrusion detection and prevention system monitors the network traffic, typically with a network adapter running in promiscuous mode and analyzing the network traffic in real-time[15]. A host-based intrusion detection and prevention system has as its purpose to protect the single host it is installed upon. This device looks for attack and malicious activity on a single host, and analyzes the network traffic to and from the computer together with possibly monitoring processes, logs and activity on the host. Common for the different types of intrusion detection and prevention systems are, which type of detection model they use. The different detection models will be explained briefly.

Anomaly detection Anomaly detection is the technique of establishing a profile of normal user behavior and comparing this profile to the actual user behavior. Any deviations from the normal behavior raise an alert or reaction. A difficulty with anomaly detection is that it is very difficult to separate what is normal and what is considers malicious. Often can malicious activity be camouflage as normal activity, and thus avoid anomaly detection[23].

Misuse detection Misuse detection is when a system uses known signatures when looking for intrusions. Much like Anti-Virus software, will misuse detection continually need to update the signature base in order to recognize what is considered to be malicious activity[23].

[2] `http://www.spywarewarrior.com/rogue_anti-spyware.htm`

2.3.5 Security suite

The Merriam-Webster Online Dictionary[3] defines the word *suite* as "a set of computer programs designed to work together and usually sold as a single unit". These collections of programs often share a common user interface and the ability to interact with each other. Such software suites are also the security manufacturer providing to its customers. These security solutions are made to fully protect the users PC, and are sold in a large variety of products. For home PC users there are several different security suites available, with all the major anti-virus companies having their own selection of security suites. These security suites includes firewall, anti-virus, anti-spyware to mention some of the most basic security features, into one comprehensive security package.

One can find a vast variety of different security suites, providing some differences in what kind of security mechanisms included. Some vendors provide several different suites, where some include only the most basic security mechanisms and others include all the security mechanisms offered by the vendor (this often includes the possibility of secure online backup etc.). One such example can be Symantec's Norton products, where Norton Internet Security (NIS) consists of all mechanisms provided to secure a computer from online threats and Norton 360 consist of all those mechanisms that NIS contains including secure online backup and some PC tune-up mechanisms.

Some examples of security suites available for the home PC marked are.

- AVG Internet Security
- BitDefender Internet Security
- BullGuard Internet Security
- CA Internet Security
- ESET Smart Security
- F-Secure Internet Security
- Kaspersky Internet Security
- Norman Security Suite
- Norton Internet Security and Norton 360
- McAfee Internet Security
- Panda Internet Security
- SOPHOS Security Suite
- Steganos Internet Security
- Trend Micro Internet Security and Internet Security Pro
- Windows Live OneCare
- ZoneAlarm Internet Security

[3] `http://www.merriam-webster.com/`

2.3.6 Online security scans and virus detections

There exist several online sites, which allow users uploading a file to scan it for viruses and malware. Virustotal[4] is one such service, which utilizes multiple anti-virus engines when searching the file. At the time of writing, are there 33 companies that participate with their anti-virus engines. Another similar site is the Jotti's malware scan[5], which utilizes at the time of writing 20 anti-virus engines. Virus.org also provides a malware scanning service, called Virus.Org Rogue File Scanning Service[6], which utilizes at the time of writing 21 different engines.

Many other services online also provides some form of security scanning, either by providing free virus scan of a computer, scanning for security threats available or file scanners as the ones mentioned above. These online virus scans uses Microsoft ActiveX technology to scan for malicious code on the users computer, and can be used as a reactive security measure. Most of the major anti-virus companies also provides free online virus scanners, some examples are listed below.

- Trend Micro HouseCall - `http://housecall.trendmicro.com/`
- F-Secure Online Scanner - `http://support.f-secure.com/enu/home/ols.shtml`
- Symantec Security Check - `http://security.symantec.com/sscv6/default.asp?langid=ie\&venid=sym`
- Panda Active Scan - `http://www.pandasecurity.com/activescan/index/`
- Kaspersky Online Scanner - `http://www.kaspersky.com/virusscanner`
- McAfee FreeScan - `http://www.mcafee.com/freescan`
- CA (eTrust) Virus Scanner - `http://www.ca.com/us/securityadvisor/virusinfo/scan.aspx`

[4] `http://www.virustotal.com/`
[5] `http://virusscan.jotti.org/`
[6] `http://scanner.virus.org`

3 Related work

In this section the related work that has been identified to suite this thesis will be explained briefly. There will be a section on security testing, with respect on how to test this thesis' relevant computer protection mechanisms (explained in Section 2.3). There will be a section on usability engineering, where the fundamental aspects of usability will be explained together with theory, methods and how to measure this. Then there will be a section on current vulnerabilities and threats, and malicious software (malware) that often affects home computer security will be explained. It is important to understand the knowledge available on the different subjects, to be able to conduct and understand this thesis.

3.1 Security testing

There exist several different types of software testing, one important aspect of this is the security test. This test is performed to find vulnerabilities of a system, and can be carried out in a great variety of different ways. There exist several different testing methods such as *White Box* and *Black Box* testing, where the differences of these two techniques is the perspective of the test. With black box testing, the most common testing methodology, is there no knowledge about the test object's internal structure. Common for most kinds of security testing is the main security concepts that they are designed to cover, such as (but not limited to) confidentiality, integrity and availability.

A way of measuring security products effectiveness is to perform security testing on these products. For this you need security testing methodologies and manuals, one such is the Open-Source Security Testing Methodology Manual (OSSTMM), created by Pete Herzog[24]. This methodology and manual in security testing is made to make security testing a measurable value, and serves as a basis for operational security auditing. There also exist several guidelines on security testing and network security testing, for instance the National Institute of Standards and Technology (NIST) special publication 800-42[25]. Another common way of evaluating security is through the framework that the Common Criteria (CC) provides. The Common Criteria[26] is an international initiative to develop criteria for evaluation of IT security, CC is focusing more on the process than the actual product and the evaluation results are divided into seven assurance levels achieved that are called Evaluation Assurance Level (EAL).

Measuring effectiveness of security products would help to evaluate different products up against each other. This could have a major significance in deciding what product to choose when having several choices. The science of security testing is a wide and difficult area, it depends very much on the application that is to be tested. When testing anti-malware products are there required significant time, knowledge and resources. There are several different test methodologies for testing anti-malware, which will be explained more in detail in Section 3.1.1. Unfortunately does there exists several controversies [27, 28, 29, 30] when measuring the security effectiveness of anti-malware products, for instance with different methodologies and testing techniques.

One reason for this can be that there does not exist an anti-malware testing standard (yet). In early 2008 were an organization established to defeat this problem, the Anti-Malware Testing Standard Organization[1] (AMTSO). This organization consists of more than 40 security software technologists, and is dedicated to helping improve the objectivity, quality and relevance of anti-malware technology testing. In the future this organization will provide a forum for discussions related to testing anti-malware, develop objective standards and best practices for testing anti-malware products and provide analysis and review of current and future anti-malware testing.

To be able to measure the effectiveness of security suites (before AMTSO releases a standard), which incorporates several security mechanisms into one comprehensive package, should one be aware of how the different mechanisms are security tested.

3.1.1 Anti-malware testing

One way of testing anti-malware products without using real malware, is by using the "EICAR Standard Anti-Virus Test File"[2]. This is a test file that makes most anti-virus products react as it were a virus, but is in fact not. Using such a test file, makes sure that the anti-malware product works as it should.

Another way of testing anti-malware products is to provide real malware to the product in a proper environment. This is the most common and correct way of testing these kinds of product. Such an approach can for instance test on-demand detection rates, malware removal[31], proactive and reactive detection. To test a product against real malware, one needs to know what is considered as real malware. Malware that have been reported spreading in the real world, and not being designed for a specific test meets this criterion. In 1993 started Joe Wells collecting reports of which viruses were found in the real world, this list become known as the *WildList*. Viruses appearing on this list, or *In The Wild (ITW)*, are only does viruses reported from several different sources. The contributors to the WildList are mostly made up by those working within the anti-virus (also known as the anti-malware) industry. In [32] is there mentioned several problems with the WildList.

In [28], Andreas Marx describes a methodical framework for anti-virus program testing. This framework is written for data security managers and for professional tester who write for magazines. This paper explains the differences between black box and white box testing, where black box tests are exterior tests without knowledge about the internal structure of how the program works. There are also different ways of testing anti-malware, with detection score, are programs evaluated on how they detect different malware. Tests can focus on the disinfection rates, where the programs ability to successfully remove malware is tested. Performance test, on-demand test, memory detection tests, false positive tests, unknown virus detection and many other tests exists as well.

One way of testing an anti-malware program's ability to detect unknown threats is to perform what is called a retrospective test [33]. The main idea about such test, is to take for instance a three-month-old scanner and compare detection rates over the malware that appears within the last three months. This way can possibly measure the products pure capability to detect new viruses, but is difficult to perform and validate.

[1] `http://www.amtso.org/`
[2] `http://www.eicar.org/anti_virus_test_file.htm`

In [28] there are also described several steps of anti-malware testing, which will be explained below.

Getting a malware collection

To be able to test an anti-malware product, is there a need for a malware collection. This collection should ideally contain all kinds of significantly different malware (see Section 3.4) and widely spread malware. Unfortunately (or maybe not) is such a malware sample set extremely difficult to get your hands on. The researcher, vendors and other collectors which are in possession of such comprehensive malware samples, are very careful with it. This because, a large malware sample set can cause quite a lot of damage and a golden rule in the anti-virus community[3] is that malware samples should not at any cause be further distributed. Another golden rule of the anti-virus community, is that malware should never be created, not even for testing purposes. There exist several malware construction kits, which have the ability to write new variants of malware. The anti-virus community firmly believes that such a method should not under any circumstances be used, because there exist enough malware as there is. Andreas Marx describes the possibility to ask the anti-virus companies to supply malware, however he points out that this will fail in most cases because the companies are very reluctant to give out sample sets. If one were so lucky to get such a sample set, is there also some other problems with such a sample set. This sample set would greatly increase the performance of that vendor's product, and can give very unfair evaluations. This is why one also should be very careful of "sponsored" tests, which often favors some products.

Another source for getting a malware sample set is to download them from the Internet, either from a special virus exchange or ftp site, or from bulletin board systems. By using this method, should one be aware of that the sample set might contain non-viral programs as well. Therefore can for instance the detection rates vary quite a bit, depending on the non-viral samples being detected or not. Some products might not detected them, because they are not looked upon as dangerous, while other products do detect them (possibly in a falsely matter).

Sorting a malware collection

Andreas Marx further describes that after a malware collection has been gathered, is there the need for sorting and cleaning the sample set. This to avoid possible non-viral samples being marked as viral samples, and to make sure that the sample set can be representative. The samples should also be verified to actually be malware and which category it belongs to, this can be for instance be done by individual analysis (such as for instance reverse engineering) or by the report files from the anti-virus program. One problem with the latter approach, is that different scanners can (and very often does) have different names and categories schemes. Another problem might be that the viral sample has changed itself, so that further analysis of it is required. With sorting the malware sample set, should one also take into account balancing the weight for the different categories. This is important for deciding what part of the anti-malware product that is the most important for the evaluation. A possibility is to weight the different samples, but the samples alone should not decide whether the product is good or bad.

[3] This community is best known as the anti-virus community, but should actually be called anti-malware community.

Problems with anti-malware testing

There exist several problems with anti-malware testing. In the article by Joe Wells [30], he points out that good anti-virus testing requires good input, good testers and good focus. With bad input, or bad sample sets can evaluations give a wrong picture and draw false conclusions. With wrong focus, can the important aspects be neglected or given the wrong attention and with bad testers can all of the above happen. It is far more likely that the sample set is bad, rather than the product.

In [27], the problem of choosing right test-set is further explained. The size of the collection should be of some magnitude (test-set containing millions of malware samples are not unheard of), the test-set should be well-ordered and maintained. Test suites should be dynamic, as malware is dynamic. The problem of biased evaluations and evaluator is discussed, where the problem of obtaining sample sets from specific vendors are mentioned. Ensuring to test different aspects in a correct and effective manor, and other problems are mentioned as well.

In [29], Igor Muttik explains the problems of unfair tests, were "random pick" with small and large sample sets can influence the results in many sorts of ways. The size of the sample set should include as many samples as possible. Problems with retrospective testing are how to determine the date of birth for some malware. A malware may have appeared for some time before it gets "publicly" known, and therefore only be detected by some vendors.

In the article[34] from 2004, the lack of testing techniques designed for testing malware detectors and problems of testing them are described. In this paper the use of program obfuscation to generate test cases from known malware is presented. Obfuscation technique, briefly explained in Section 3.4.1, is a technique which malware creators often uses when trying to make their creations undetectable by malware scanners. This technique was quite successful, and the early test shows that commercial scanners were then not resilient to common obfuscation transformations. With commercial malware detectors, being ever more sophisticated and utilizes different detection techniques, are more obfuscated malware detected. From [35], the change in recent years in virus research shifting from syntax based signature generation to semantics based signature generation, can common signatures better detect obfuscated malware. Furthermore have there been developed other ways of detecting obfuscated viruses, such as the similarity analysis.

3.1.2 Firewall testing

The main goal of a firewall is to monitor and close open ports. There exist 65 535 ports on a computer, which the firewall has to protect. A simple way of testing the firewall is to perform a port scan, to see which ports are open or closed. A firewall can also "stealth" a port, in order for it to not appear on for instance remote port scans[4]. As one might know, is that a traditional firewall only is as good as its rules. If a rule in the firewall, allows through connections to the FTP (File Transfer Protocol) port 21, are all FTP connections to this port sent through the firewall without any further checks to see if the connection is valid and proper. Newer and more advanced firewalls can sometimes check for faults in the FTP connection as well, but it still relies on its rules. So a faulty rule set can seriously affect a computer's security. With personal firewalls, were

[4]A popular non-malicious remote port scanning service is ShieldsUp! available at: www.grc.com

default rule set are provided by the vendor, is it up to the users to add their own rules when needed (often via a pop-up from the firewall, asking the users whether to accept or deny a connection through the firewall).

Another way of testing the security capabilities of a firewall, is to perform a so called "leak test"[36]. Leak tests[5] are small legitimate programs, with its only purpose to test different capabilities of a firewall and report it to the users. The first well known leak test, were made by Steve Gibson. Different leak tests, evaluates the different aspects of a firewall. They can for instance test a firewall for outbound protection, DLL injection and many other firewall vulnerabilities which are commonly used by malware. A problem with some common leak tests are that some vendors have been known to cheat, by detecting the actual leak test file (so it seems that the leak is not existing) without detecting other malware exploiting that same leak.

Matousec[6] is a small group of people which specializes in security software testing. Their main goal is to improve security of end-users with their security related projects and research. One such project is their Firewall Challenge, which methodological puts several different security products firewalls to the test. This site has put together a Security Software Testing Suite (SSTS), which basically is a set of tools used for testing personal firewalls and Internet security suites for Windows. It is based on the idea of leak tests, small independent programs that attempt to bypass an outbound protection of the security software. SSTS also consists of many independent programs that test specific features.

3.1.3 Independent security test laboratories

There exist several independent security testing laboratories, which evaluates and certifies anti-malware products. Some of these commonly known and widely accepted certification organizations are, the ICSA Labs Certification[7], West Coast Labs (WCL) Checkmark Certification[8] and Virus Bulletin's VB100 award[9]. All these certification organizations evaluates anti-malware products, and requires the malware detectors to identify all ITW malware with a detection rate of 100% for passing their certification.

There also exists some respected online independent anti-virus software testers, namely Andreas Clementi's AV-comparatives.org and the German institute AV-test.org. Both these security testing laboratories are well known within the anti-virus industry, and conducts regularly tests on different security solutions.

3.2 Usability

In the field of usability, usability engineering, human-computer interaction (HCI), user-centered design (UCD), man-machine interface (MMI), ergonomics or what you would like to call it (there are some subtle differences between some of the terms, but this would not be a part of this thesis), is it done a lot of research over several decades. With computers becoming more commonly used by all kinds of human beings in all kinds of situations, have these different subjects become

[5] Several leak tests can be found on: `http://www.firewallleaktester.com/index.html`
[6] `http://www.matousec.com/`
[7] `http://www.icsa.net/icsa/icsahome.php`
[8] `http://www.westcoastlabs.org/`
[9] `http://www.virusbtn.com/vb100/index`

a very important part of the computer industry.

Jakob Nielsen in [37] describes usability as a part of a more general concept of *system acceptability*, this concept is defines as follows:

> "... the question of whether the system is good enough to satisfy all the needs and requirements of the users and other potential stakeholders ...".

This relation can be illustrated, as defined by Nielsen, in Figure 2.

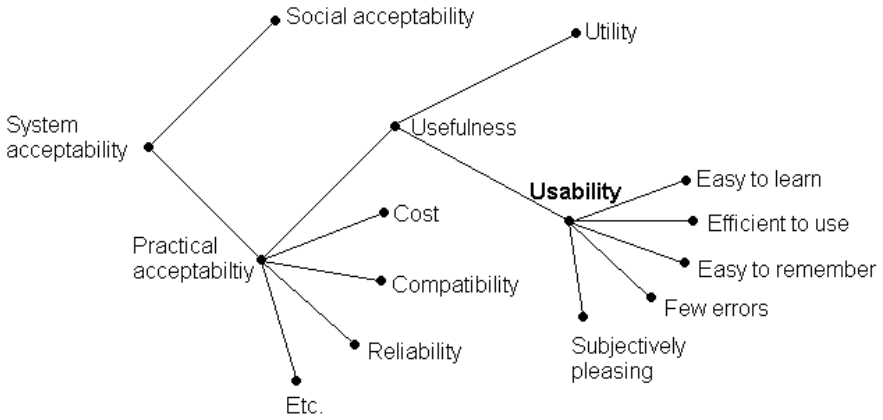

Figure 2: A model of the attributes of system acceptability (or in other words Nielsen's *system acceptability* taxonomy), taken from [37].

In Nielsen's model, overall acceptability of a computer system is a combination of *social acceptability* and *practical acceptability*. An example of a system that might not be socially accepted, can be a system that is offensive to certain subjects. Given that a system is socially acceptable, can usability be explained as a part of practical acceptability. Practical acceptability includes traditional categories such as cost, compatibility, reliability and usefulness. Usefulness is the issue of whether the system can be used to achieve some desired goal, and are further divided into the two categories of *utility* and *usability*. *Utility* is defined as the functionality of the system and whether it can in principle do what is needed. *Usability* is defined as the question of how well users can use that functionality, and can be further divided into five usability attributes.

- Learnability - Refers to how easy the system is to learn for the user, in order to effectively achieve useful work.
- Efficiency - Refers to how efficient the system is to use, in order to get a high level of productivity once the user has learned the system.
- Memorability - Refers to how easy the system is to remember, in order for casual users to be able to return to the system without having to learn everything all over again.
- Errors - Refers to the error rate of the system, so that the users make few error during the use of the system. If the users do make errors they need to easily recover from them.
- Satisfaction - Refers to how pleasant the system is to use, so that the users are subjectively

satisfied when using it.

In [37], another important issue for usability are the users' task and their individual characteristics and differences. Nielsen describes an important aspect of usability engineering, namely to know the user. Understanding the major ways of classifying users may help when addressing usability. There are three main dimensions along which users' experience differs, as shown in Figure 3; experience with the system, with computers in general and with the task domain.

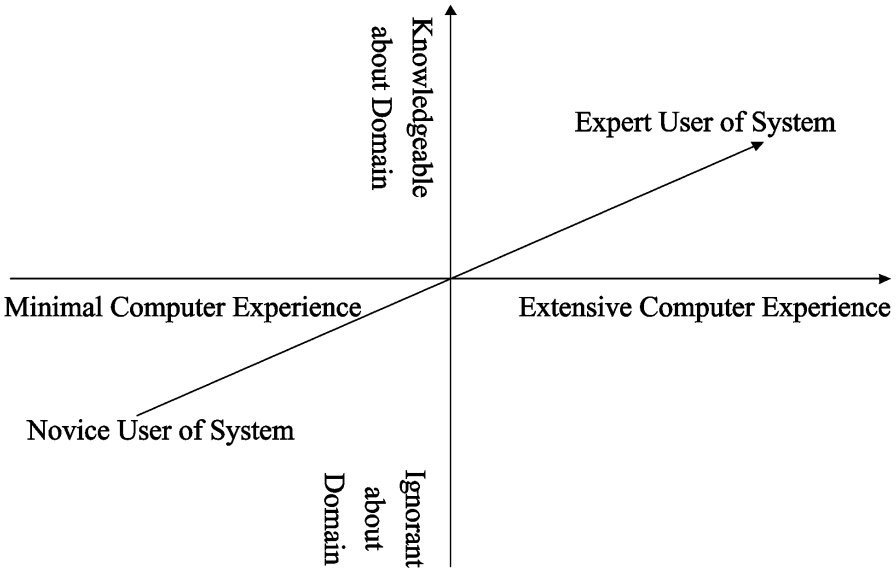

Figure 3: The three main dimensions on which users' experience differs: knowledge about computers in general, expertise in using the specific system, and understanding of the task domain, taken from [37].

3.2.1 Standards

There have been developed several standards addressing the aspect of people working with computers, like for instance the ISO 9241[10] and ISO 20282[11] standard[38]. The ISO 9241-11[39], which is a part of the ISO 9241 standard, defines usability as:

> "Usability: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." - Taken from [38].

To fully understand this definition, must some of the key elements be described.

- Effectiveness - The user's ability to complete relevant tasks using the system in order to achieve some specified goals.
- Efficiency - The amount of resources consumed in performing the tasks, both physical and cognitive workload.
- Satisfaction - User's subjective reaction to using the system.

[10] ISO 9241: Ergonomic requirements for office work with visual display terminals (parts 1 to 17)
[11] ISO 20282: Ease of operation of everyday products

These key components of usability, mentioned above, are a part of the ISO 9241-11 standard framework for specifying usability and the relationship between the components are illustrated in Figure 4. This figure shows what information is needed when specifying or measuring usability.

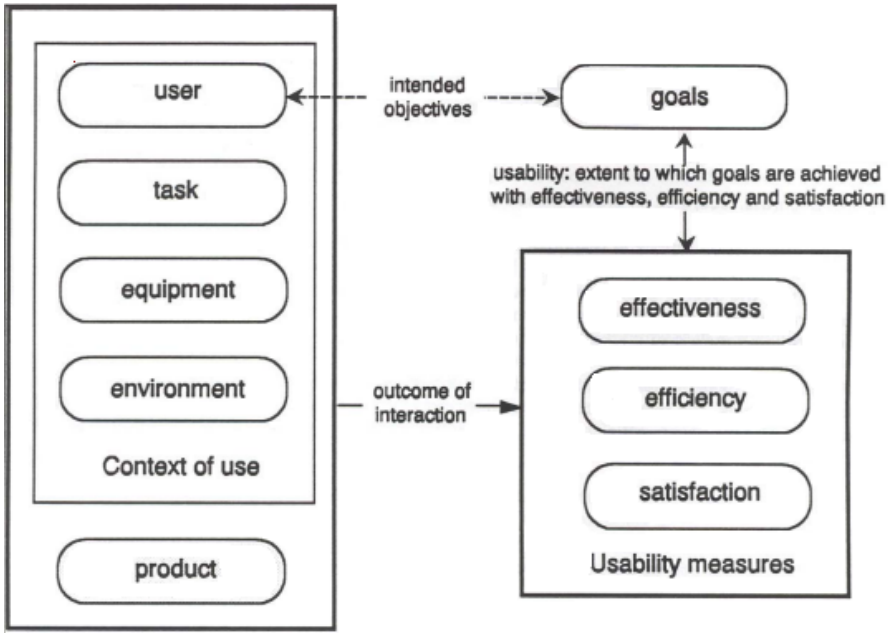

Figure 4: Usability framework, taken from [39].

As stated by the ISO 9241-11 standard, as shown in Figure 4, is it needed to describe the desired goals of the use of a product. Furthermore is it needed to describe the context of use, which includes a description of the users, tasks, equipment and environment. This framework helps specifying usability of products and can be of great help when addressing the issues of usability.

3.2.2 Usability testing

Usability testing [40] is a technique that involves testing the system in question on end-users. Testing on real users can give designers and developers valuable input on how the users use, likes and performs on the system. With this input, can possible errors and areas of improvement be discovered. Such usability testing is often done in a planned manor throughout the production period of a given system, for instance in an iterative order, to ensure that possible user problems are discovered and removed. By properly testing the system against real end-users, is it a much larger possibility that the system can become a success, compared to a system which has not performed usability testing.

Usability testing can be performed with a great variety of methods. From strict empirical usability testing on a systems actual users for evaluating a product, to more of an inquiring method were users are interviewed or expected to comment on a system in general. Furthermore to different usability inspection methods incorporating end-users.

3.2.3 Inspection methods

Usability inspection, as explained by Nielsen in [37, 41], is a set of different methods that are based on having evaluators inspects a user interface. These evaluators can be usability experts, designers (other than those who have made the interface of course), domain experts or just normal users (which sometimes are referred to as user experts). The inspections aim is mainly to find usability problems within a design, but can also address the overall usability of an entire system.

There exist several types of different inspection methods. In [41], the following usability inspection methods are described.

- Heuristic evaluation
- Heuristic estimation
- Cognitive walkthrough
- Pluralistic walkthrough
- Feature inspection
- Consistency inspection
- Standards inspection
- Formal usability inspection

Were maybe the two most famous usability inspection methods are, *heuristic evaluation* and *cognitive walkthrough*. These two methods will be described more in detail.

Heuristic Evaluation

Heuristic evaluation is a method developed by Jakob Nielsen together with Rolf Molich in 1990[42, 43]. This method is an informal method of usability analysis, where a small number of evaluators looks at an user interface design and judges its compliance with recognized usability principles (the "heuristics").

Heuristic evaluation is known as a "discount" usability evaluation method, meaning that its goal is to locate major usability problems in a system without using the large amount of resources typically required for usability testing. This discount approach uses simpler and more approximate methods, than the more formal and exact methods of testing. Heuristic evaluation is performed by having each individual evaluator inspect the interface alone, in order to ensure independent and unbiased performance. Furthermore can an evaluation be recorded either as written reports or by having the evaluators verbalize their comments to an observer. Using an observer (also called the "experimenter") in a heuristic evaluation, are different than using an observer in a traditional user test. In traditional user testing, observers are not as willing to answer questions from the evaluators during the sessions and to provide hints. This is because that in traditional user testing one normally wants to discover the mistakes users make when using the interface. While in heuristic evaluation, and especially in domain-specific applications, are the experimenter much more allowed to "help" the evaluators, particularly if nondomain experts are serving as evaluators. Answering the evaluators question will enable them to better assess

the usability, claims Nielsen. Providing the evaluators with hints if they are somewhat stuck, also ensures not to waste precious evaluation time. If of course, the reason why the evaluator is stuck and the usability problem in questioning are commented on[37].

A general recommendation when performing an interface inspection, is to let the evaluators get a feel for the system, before they starts to evaluate the usability of it. Heuristic evaluations aims at explaining each observed usability problem with reference to an established usability principle (see Section 3.2.4 for more information), and does not provide a systematic way of fixing usability problems or assess the possible quality of any redesign schemes. When performing a heuristic evaluation, Nielsen recommends that normally should there be at least three to five evaluators. Using smaller number of evaluators and problems might be overseen, and by using a greater number of evaluators one does not gain that much additional information. Figure 5, shows the proportion of usability problems found compared to the number of evaluators.

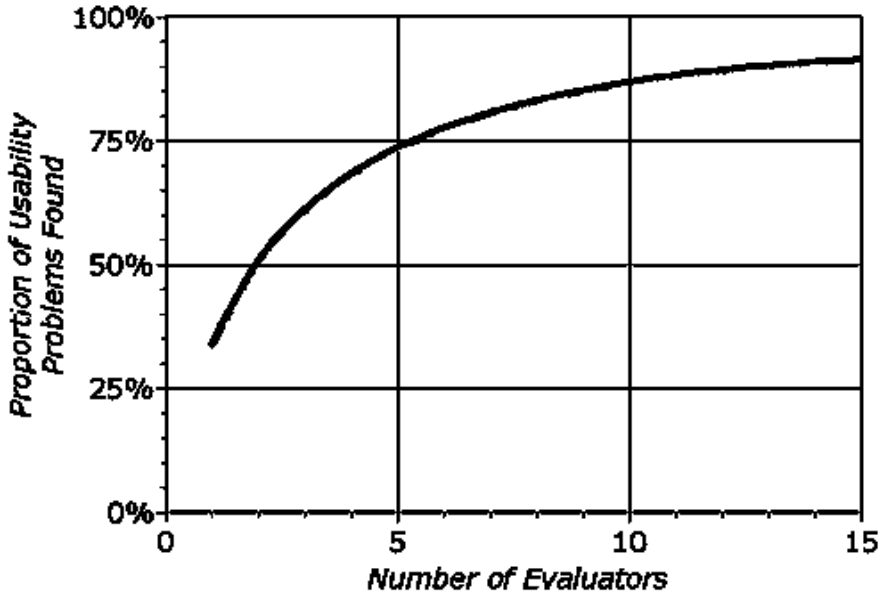

Figure 5: A curve showing the proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators. The curve represents the average of six case studies of heuristic evaluation, taken from [41].

A note made by the creator of this evaluation method, Jakob Nielsen, is that this technique may produce discount results. Nielsen and other usability experts advocate using heuristic evaluation as a supplement to usability testing.

Cognitive walkthrough

Cognitive walkthrough [41] originates from the software engineering technique called code walkthrough. Code walkthrough is when a segment of program code is reviewed sequentially with respect to check for certain characteristics, like e.g. coding and convention style are adhered to. In cognitive walkthrough, a sequence of actions the users needs to perform to accom-

plish some tasks, are checked for potential usability problems. The main focus of a cognitive walkthrough is to establish how easy a system is to learn, and is based on the notion of learning through exploration.

A common cognitive walkthrough can be divided into two phases, a preparatory phase and an analysis phase. In the preparatory phase the experiment creator needs to determine the interface to be used, its likely users, the task and the actions to be taken during the task. In the analysis phase the evaluators goes through four steps of human-computer interaction, developed by Polson et. al.[44].

1. The user sets a goal to be completed within the system.
2. The user determines the currently available actions.
3. The user selects the actions that they think will take them closer to their goal.
4. The user performs the action and evaluates the feedback given by the system.

The famous article by Whitten and Tygar "Why Johnny Can't Encrypt"[45], uses cognitive walkthrough to asses the usability of the secure e-mailing system called Pretty Good Privacy (PGP) 5.0, together with an user test. This article shows that cognitive walkthrough can successfully be used to asses the usability of a system. Furthermore shows this article that PGP 5.0 were not sufficient to secure e-mail, because it were not usably secure for people who are not already knowledgeable in that area. With only one third of the participants able to correctly sign and encrypt an e-mail message with PGP 5.0, even though the participants were generally educated and experienced at using e-mail.

3.2.4 Usability principles

There exist several usability principles that, if followed, greatly enhances the possibility of making a system usable and user-friendly. One such famous principle first and foremost coined for security, are Saltzer and Schroeder's principle of *psychological acceptability*.

> "Psychological acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors."
>
> -Saltzer and Schroeder [46].

In [47], the following principles are described.

- Structure - Organize a user interface purposefully, in meaningful and useful ways that put related things together and separate unrelated things.
- Simplicity - Make common tasks simple to do and communicate simply in the user's own language.
- Visibility - Keep all needed options and materials for a given task visible without distracting the user with extraneous or redundant information.
- Feedback - Keep users informed of actions, changes of state or condition and errors or exceptions using clear, concise and language familiar to the users.

- Tolerance - Be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing, while preventing errors wherever possible by tolerating varied inputs and sequences.
- Reuse - Reduce the need for users to rethink and remember by reusing internal and external components and behaviors and maintaining consistency with purpose.

In [40], Jeffrey Rubin explains the following four factors as key elements of any generally accepted usability definition.

1. Usefulness - This concerns the degree to which a product enables a user to achieve his or hers goals, and is an assessment of the user's motivation for using the product at all.
2. Effectiveness (ease of use) - This element is usually defined quantitatively, either by speed of performance or error rate, and is tied to some percentage of total users.
3. Learnability - This has to do with the user's ability to operate the system to some defined level of competence after some amount of training. It can also refer to the ability of infrequent users to relearn the system after periods of inactivity.
4. Attitude (likability) - This refers to the user's perception, feelings, and opinions of the product, usually captured through both written and oral interrogation.

Nielsen and Molich [42, 43] came up with a set of usability heuristics in 1990, which Nielsen later refined in [41]. In Table 1 these heuristics or principles can be seen.

Original heuristics	Refined heuristics	Description
Simple and natural dialogue	Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information competes with the relevant units of information and diminishes their relative visibility.
Speak the user's language	Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
Minimize user memory load	Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.
Be consistent	Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing.
Provide feedback	Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Provide clearly marked exists	User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Provide shortcuts	Flexibility and efficiency of use	Accelerators (unseen by the novice user) may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.
Good error messages	Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Prevent errors	Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and prevent users with a confirmation option before they commit to the action.
	Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Table 1: Nielsen's and Molich original heuristics and the matching refined heuristics, together with a short description by Nielsen of the heuristic.

Usability principles that many open source projects follow, are the GNOME Human Interface Guidelines[12]. This guideline includes the following principles.

- Design for People
- Don't Limit Your User Base
- Create a Match Between Your Application and the Real World
- Make Your Application Consistent
- Keep the User Informed
- Keep It Simple and Pretty
- Put the User in Control
- Forgive the User
- Provide Direct Manipulation

3.2.5 Measures of usability

The ISO 9241-11 standard usability framework (see Figure 4), mentions what information is needed for measuring usability. With this in mind, can usability testing or reviews be designed to ensure an appropriate and valid measure of usability with regards to the context of use. In Table 2, one can see examples of usability measures.

Usability objective	**Effectiveness measures**	**Efficiency measures**	**Satisfaction measures**
Overall usability	Percentage of goals achieved; Percentage of users successfully completing task; Average accuracy of completed tasks	Time to complete a task; Tasks completed per unit time; Monetary costs of performing the task	Rating scale for satisfaction; Frequency of discretionary use; Frequency of complaints

Table 2: Examples of measures of usability, taken from ISO 9241-11[39]

In [48], Bevan et. al. divides measuring the cognitive workload[13] into an objective and a subjective measure. Objective measures are independent of personal judgments related to task complexity and are not directly under the conscious control of the subject. One example of such a measure, can be the subjects heart rate variability. With objective measures one can get (to a certain degree) an unobtrusive measure taken during the actual task performance. Subjective measures on the other hand are related to for instance questionnaires performed after a task. Here the subjects can consciously control the outcome of the evaluation.

[12] `http://library.gnome.org/devel/hig-book/stable/`
[13] Cognitive workload relates to the mental effort required to perform tasks[48].

Examples of subjective measures, are the Software Usability Measurement Inventory (SUMI)[14] questionnaire developed by the University College Cork. This is a 50-item post-task questionnaire that measures user satisfaction. A SUMI questionnaire takes approximately 10 minutes to complete and needs at least 10 representative users to get accurate results. Another example of a subjective measure are John Brooke's System Usability Scale (SUS) [49], which is a simple 10-item post-task questionnaire. John Brooke describes this method as a "quick and dirty" usability scale, which allows for a low cost assessment of usability in systems evaluation.

3.3 Vulnerabilities and threats

A good way of getting a quick insight into the different threats and vulnerabilities are through taxonomies. There exist several different categories of vulnerabilities and threats to a personal computer, which will be briefly further explained here.

3.3.1 Types of vulnerabilities

The book of Rick Lehtinen et al. [1] categorizes the vulnerabilities in computer security to the categories described below.

Physical vulnerabilities Physical security is related to securing the physical equipment and environment where the equipment is kept. Locks, alarms and other measures are important first defense against break-ins.

Natural vulnerabilities Natural vulnerabilities are related to natural disasters and other accidents like earthquakes, floods, lightning, fire, etc. These kinds of vulnerabilities can often be difficult to prevent, but one should be aware of them and the risk it presents.

Hardware and software vulnerabilities Hardware vulnerabilities can happen if certain kind of hardware failure leads to a compromised system. Other hardware vulnerabilities can be when introducing special hardware, which open security holes or fails the protection mechanism. Software vulnerabilities are software failures of any kind that may cause your system to fail, become unreliable, open for attacks or simply can not be trusted. Software errors can occur because of bugs in the design, implementation, installation or configuration. Malware often relies on software vulnerabilities to spread and cause havoc in a system.

Media vulnerabilities Media vulnerability is associated with the risk involved in handling and storing information on any type of media. For instance can backup media such as CD, DVD, printout, etc. be stolen or damaged, and thus become a vulnerability if not proper cared for.

Emanation vulnerabilities All electrical equipment emits electromagnetic radiation, which can be intercepted by a skilled adversary. These types of vulnerabilities are often of somewhat a military or high security profile, where the information is of critical importance.

Communications vulnerabilities Communication vulnerabilities are when a computer is connected to a network or some other instances like for instance the Internet. Packages can be

[14] SUMI Questionnaire Homepage: `http://sumi.ucc.ie/`

intercepted, misrouted and forged in a communication network. Wireless communication is also included into this category.

Human vulnerabilities The last vulnerability category that Rick Lehtinen et al. [1] mentions, are maybe the biggest vulnerability of them all, the human vulnerability. Human vulnerability includes every aspect from human errors to social engineering, where humans are tricked into for instance giving away their password.

Gary McGraw in [50], claims that typical software vulnerabilities falls into two categories, bugs at the implementation level and flaws at the design level.

3.3.2 Types of threats

Furthermore does the book by Rick Lehtinen et al. [1], Computer Security Basics, categorize threats into three main categories.

Natural and physical threats Natural and physical threats imperil every aspect of computer security, often can one do very little against it. It is therefore important to be aware of them in order to take the precautions necessary, like having backup of critical data.

Unintentional threats The unintentional threats are a difficult category to avoid, because they happen by accident. Lack of proper training and knowledge together with ignorance are key aspects in unintentional threats, and this kind of threat often causes more compromised data than intended and malicious threats.

Intentional threats Intentional threats are direct attacks that are feasible by certain types of attackers. Intentional threats are often categorized into two varieties, outsiders and insiders. In home computer setting are the insider threat not such a menace, and often ignored. If the home network, consists of a wireless network that is not properly secured together with the possibility that some with malicious intents are near by, can insider threat be a menace that needs to be considered. In large corporations on the other hand, poses insiders a real threat that should be considered.

3.3.3 Threat vs. Attack

In [15], the different classes of attacks that posses as a threat against computer systems are the following.

Stealing Passwords The easiest way into a computer, or maybe a users Internet bank, social network, etc. are usually through the front door, which in computer terms is the *login* command. Nearly all systems rely on a form of login system, where a user supplies a login name and password. A big problem with this system is that people tends to pick very bad passwords, or stores the password in an insecure place. Therefore can password stealing be a major threat for a system.

Social Engineering This threat is maybe one of the most common and successful threats being used in recent years. Either it is via e-mail, instant messaging, or any other way, social engineering tries to manipulate people into giving away confidential information like your password or credit card details.

Bugs and Backdoors Bugs, and potentially backdoors, are a huge problem in computer science. Writing correct software, that not contains any dangerous bugs that can be exploited to for instance create backdoors to a system, seems to be a problem that never disappears.

Authentication Failures Failure of authentication mechanisms are not always because of users pick poor passwords or write them down in an insecure place. Sometimes can authentication mechanisms be defeated without it being the users fault. Poorly designed authentication mechanisms that are subjected to eavesdroppers or man-in-the-middle attacks, can just as easy defeat the trustworthiness of the mechanisms.

Protocol Failures Similarly to authentication failures, can protocols be inadequate or buggy. A good example of this is the wireless data communication protocol WEP, that were designed for providing confidentiality comparable to that of a traditional wired network (were eavesdropping is much more difficult).

Information Leakage Computer systems often leak some form of information that can be used against that system. Malicious hackers and social engineers can cleverly use information that a system leaks, to successfully achieve their goals. This can for instance be a port scan or DNS query.

Exponential Attacks - Viruses and Worms These kinds of attacks uses programs like viruses and worms to spread themselves, and succeeds by exploiting common vulnerabilities.

Denial-of-Service Attacks These kinds of attack wants to prevent legitimate use of a service, by simply overusing the service. This overuse strains software, hardware or network links beyond their intended capacity, and thus leaves a service unusable. Denial-of-Service (DoS) attacks are when a single host shut down or degrade the quality of a service, while Distributed Denial-of Service (DDoS) are when many hosts attack a service.

Botnets Botnets are groups of bots (or robots, zombies etc.), that is computers which have been compromised and contains some malicious program that allows it to be remotely controlled. These groups can be used for a variety of malicious purposes, like for instance participating in a DDoS attack, spreading SPAM, malware and other malicious activities.

Active Attacks These kinds of attacks are when an attacker is actively attempting to cause harm to a network or system. This type of attack requires the attacker to be positioned in way so that he or she can communicate between the victims.

3.3.4 Phishing and Pharming

Gunter Ollman describes the threat of phishing in [51], as one of the greatest 21st century crime. Phishing uses social engineering to trick the users into imparting their confidential information for nefarious use. The victims of phishing is lured into traps specifically designed to steal their electronic identity. One common phishing scam is a spoofed e-mail message, saying that your online banking system needs you to login to their website (which is provided as a link in the e-mail, this link of course leads you to website looking correct) and update your confidential personal information to ensure that they got the right information about you. If a user falls for this scam,

can they be a victim of identity theft, where their identity is falsely used on the Internet in a malicious way. Pharming[52] exploits well known flaws in DNS services and the way in which host names are resolved to IP addresses. This enables the attacker to redirect a website's traffic to another bogus website, which can collect confidential information. An example of pharming attack, can be by changing the *hosts* file on a victim's computer.

Other sources for information about vulnerabilities and threats

Another great source for current vulnerabilities and threats are Microsoft Security Intelligence Report[53, 54] and Symantec Internet Security Threat Report[55, 56], where you can get elaborated reports from some major companies on how the current threat activity on the Internet has been over a six-month period.

To be able to follow zero-day vulnerabilities are the electronic mailing list Bugtraq[15] a great way of doing so, but this mailing list are not well suited for the common home computer user. There also exist several websites that identifies and lists vulnerabilities and threats, for instance is Mitre's Common Vulnerabilities and Exposures[16] one such site and National Institute of Standards and Technologies (NIST) National Vulnerability Database[17] another. CVE have done work on collecting vulnerability databases, creating a dictionary of publicly known computer security vulnerabilities and exposures free for public use[57]. The security vendor Symantec has a similar site[18], listing the latest security threats. Another good website posting software vulnerabilities are SecurityFocus's Vulnerability site[19]. There also exist projects dedicated to improve security of applications, that lists vulnerabilities and threats that are of current interest. One such project is the Open Web Application Security Project (OWASP)[20], which focuses on making application security (vulnerabilities, threats, etc.) publicly available.

3.3.5 Blended threats

A term that often appears in relations with threats and vulnerabilities is the term "blended threat"[58], which is as the name suggests a combination of different threat types. A blended threat [59] is an exploit code that combines malicious code with vulnerabilities in order to quickly and successfully launch an automated attack against networked computers. These kinds of attacks often combines viruses, worms, Trojan horses and other malware, directed against known vulnerabilities to propagate. A blended threat also often incorporates malicious code that has the capability to exploit several different vulnerabilities, in order to have a greater and more probable change to succeed.

3.3.6 Summary

The most common vulnerability category affecting home computer users are *human vulnerabilities* and *software vulnerabilities*. Human vulnerabilities, including aspect such as human errors and social engineering, are difficult to avoid but with proper training and education can the risk associated with it be reduced. Software vulnerabilities can be more difficult to prevent for the

[15] `http://www.securityfocus.com/archive/1`
[16] `http://www.cve.mitre.org/`
[17] `http://nvd.nist.gov/`
[18] `http://www.symantec.com/enterprise/security_response/threatexplorer/threats.jsp`
[19] `http://www.securityfocus.com/vulnerabilities`
[20] `http://www.owasp.org/index.php/Main_Page`

common home computer users, but keeping their computers up-to-date with the latest software updates and patches greatly reduces the risk.

The threat categories most often affecting home computer users are the *unintentional threat*, this could be because of lack of proper training and knowledge as well as ignorance, and causes often unnoticed compromised data. *Intentional threats*, such as outsiders attacking a specific system, are more a threat for corporate users. Home computer users often are not summited to direct (active) attacks, but rather large scale automated attacks. Threats such as *exponential attacks* with viruses and worms, *bugs and backdoors*, *botnets* and *stealing passwords* posses as a big threat against home computers. Phishing sites stealing users passwords and identity, and botnets automated attacks against a large population are often successful because of the large number they reach out to. These kinds of attacks often combines different threat making them so-called *blended threats*.

The maybe biggest threat on the Internet is malicious software (malware), which affects corporate users as well as home computer users. In [60, 61], is the threat of drive-by downloads of malware after visiting malicious and hacked websites discussed. This threat is becoming an increasing problem, and it is introducing different malware types to many computers. To understand more about the malware problem, should one know some basics about the different categories. This will be explained in the following sections.

3.4 Malicious software (malware)

Malicious software or malware is a type of software designed specifically to disrupt a computer system (do something malicious on the computer). To be able to protect against such malware, should one understand some basics about them. In [62] Oleg Zaytsev mentions the following categories of malware.

- Computer viruses - is a self-replicating computer program that spreads by inserting copies of itself into other executable code or documents. Infecting other programs with the virus, so that the virus code is executed when the infected target is started.
- Network and mail worms - is a self-contained self-propagating program that is able to spread functional copies of itself to other computers via network and does not need any external interaction.
- Trojan Horses - a program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the program.
- Adware/Spyware - a program that have the ability to scan systems or monitor activity and relay information to other computers or locations in cyber-space. This information does not need to be “harmful”, but can contain sensitive information like credit card information and so on.

A note to these categories is that many malicious programs includes characteristics from several of the above-listed categories. For instance can a worm, include a virus in its payload and a Trojan Horse can be introduced into an application in the same manner as any other virus. As a result of this, can different anti-virus software categorize the same malware in differing ways.

In [63], IBM's Internet Security Systems X-Force team classifies malware on the most dominant feature of the threat. Malware analyzed in 2007 were divided into the following categories by the X-Force team.

- Worm - Self-propagates over a network.
- Backdoor - Provides functionality for an attacker to connect back to the victim's system without supplying authorized login credentials.
- Virus - Infects a host and does some form of damage to the host, but cannot self-propagate.
- Password Stealer (PWS) - Designed to steal the login credentials for specific online applications and is a key component in identity theft attacks.
- Downloader - Low-profile malware that exists to install itself so that it can then download and install a more sophisticated or updated malware agent.
- Keylogger - Captures all keystrokes and stores the information away for later retrieval by the attacker.
- Dialer - Uses modem connections to either dial back to the attacker or causes the victim to use primary-rate billing numbers when making connections.
- Trojan - Appears to be legitimate file before installing itself-often with rootkit functionality.
- Miscellaneous - All other malware not falling into one of the above primary categories.

One term that arises, and that many may think of as a malware category, is the term *rootkit*. A rootkit is a set of tools designed to take fundamental control of a computer system, without authorization by the system's owner. Usually rootkits subvert an operating system to avoid detection, by concealing running processes from monitoring programs, hiding files or system data from the operating system and capturing system functions or replace system libraries. One possibility why rootkit is not mentioned as a malware category as they listed above in this section, is that the technology is not malicious per se. Rootkits may be used for productive purposes, such as to solve various problems and to perform useful tasks. Malware, like Trojan Horses, often incorporates rootkit abilities to hide its present and making it undetectable by anti-malware software.

The most common malware categories, like computer viruses, worms, Trojan Horses and spyware, should be further explained in order for understanding their threat to common home computer users.

3.4.1 Computer Viruses

To be able to prevent computer viruses is it important to understand some basics on how computer viruses works and spreads. For computer viruses, as opposed to computer worms, to be spread the virus needs a host it can infect. Computer worms on the other hand is a self-replicating

computer program, that can spread (send copies of itself) without any user intervention. The virus-removal procedure, also called healing, consists of removing the malicious code of the virus from the targets body and then restoring the program's functionality. This procedure can be extremely complicated, since viruses gets exceedingly more advanced and utilizes many different obfuscation techniques in order to avoid getting detected. Such techniques can for instance be (but not limited to) encrypting some of the code, polymorphic code that mutates while keeping the original algorithm, metamorphic code that can reprogram itself too not look alike. These techniques are all trying to avoid the pattern recognition capabilities of the anti-virus software[62].

There exist several types of computer viruses, with the most common types being *file viruses*, *macro viruses*, *boot viruses* and *script viruses*. Also polymorphic viruses can be seen as separate type, because of the wide propagation of them.

Fred Cohen discusses in his PhD thesis [64], that sometimes cannot computer viruses be precisely identified and the removal of such viruses may not be easy. Cohen says that the only way of getting absolute protection from computer viruses is through absolute isolation, but this is usually an unacceptable solution. Furthermore he describes and proves that there not exists an algorithm that is perfect in detecting all kinds of viruses.

> "No infection can exist that can't be detected, and no defensive mechanism can exist that can't be infected."
>
> -Fred Cohen [64]

3.4.2 Worms

Network worms often causes great harm to the network, either by infecting a lot of computers via a network like the Internet, or by just consuming bandwidth to the point that legitimate services using the network being denied access. This can then become a Denial Of Service (DOS) attack or Distributed Denial Of Service (DDOS) if there is several computers sending network activity to one location. Healing a computer after the worm attack consists of searching and removing worm components that is left behind on the hard drive. In the case of network worms, consists the healing phase also often in protecting the vulnerability that the worm used to propagate[62].

3.4.3 Trojan Horses

There exist several different categories of Trojan Horses, if for instance the Trojan Horse's main goal is to collect and transmit confidential information about the user to the Trojan Horse's owner is it often called a Trojan-spy. Some Trojan Horses are used for allowing other malicious programs access to that infected machine, these are often called Trojan-droppers. Being infected by such a Trojan, often leads to a great deal of different malware entering the system and the machine becomes a part of a bot-net.

Trojan Horses are often introduced to a system via a downloader (often referred to as a Trojan-downloader). This small downloader secretly downloads and installs a foreign program into the system. One way of getting infected by a Trojan-downloader can be via a mail worm, which includes a small downloader in its payload. A Trojan-downloader may reside on a system, without being able to download the representative Trojan. This can be because of several factors, for instance can the site where the downloader wants to retrieve the Trojan either be offline or blocked. Or some kind of security mechanisms, like for instance an anti-virus program, can

protect the system from getting infected by the Trojan while the Trojan-downloader may slip past it guards[62].

3.4.4 Spyware

Spyware, programs that collects information about the users and passes this information to some third party without the user's consent, might be of very harmful nature but can also just be used for gathering statistics for marketing purposes. Marketing and advertising software called adware, can be very annoying with for instance pop-ups windows displaying advertising. Cookies used browsing the web, are also often categorized as spyware. Even legitimate pages with its corresponding cookies are sometimes categorized as spyware, even though they are not either harmful or used for gathering information about the users.

More harmful spyware can often incorporate hijacking capabilities, with for instance unauthorized modification of a users web browser. Often can spyware also install toolbars and other unwanted software on the computer. This can considerably slow the computers operations and generate excessive Internet traffic[62].

3.4.5 Summary

In Microsoft Security Intelligence Report[54], Microsoft report of an increase in malware removed via the Microsoft Malicious Software Removal Tool (MSRT) to over 40% during the second half of 2007. The biggest categories increasing were adware, Trojan-downloader, Trojan and potentially unwanted software.

From Symantec Internet Security Threat Report[56], Symantec report of a 136% increase in malware in the last six months of 2007 compared to the previous period. The biggest category in the last six months of 2007 were, Trojans with 71% of the volume of the top 50 potential malicious code infections. Worms came in second with 22%, and viruses and backdoors had respectively 15% and 13%.

4 Research method

This chapter will describe what method is used for performing the experiment and why.

4.1 Introduction

This thesis will deal with the usability and security of comprehensive security software products designed for personal usage, to investigate to which extend zero effort security is possible. The security software being evaluated will be some selected security suites (for more explanation on security suites, see Section 2.3.5) containing the most important mechanisms for properly protecting a home computer.

It is furthermore very important to take into consideration the knowledge and resources available, when choosing the appropriate research method too perform this thesis. With resource extensive methods, is it much more likely that this method will not be used in other settings. Therefore will this thesis emphasize on deploying resource effective methods and see how well these methods perform.

This thesis will use a qualitative research method, mainly because of the nature of the thesis. With one part being usability testing of some security products made for the home PC users and another security test of the products, would it be extremely time- and resource consuming to use a quantitative method. The empirical data gathered will be presented and analyzed as good as possible.

4.2 Usability evaluation method

In Section 3.2 is there described two usability inspection methods, *heuristic evaluation* and *cognitive walkthrough*, which were considered to be the top two alternative on how to evaluate the usability of the security suites. These two methods were singled out through a theory study and with assistance from a usability expert at Gjøvik University College.

The evaluation method that will be used for performing the usability inspection part of this thesis will be the popular and resource economical *heuristic evaluation* (see Section 3.2.3). This "discount" method is the usability evaluation method that will be used to carry out the usability experiment. In [65], Hvannberg et. al. presented a conceptual heuristic evaluation framework that involves some aspects that this thesis will follow. This includes giving the users some task they are to perform, then performing the heuristic evaluation and finally answering the post-test questionnaire [49] System Usability Scale (SUS).

4.2.1 Reliability and validity

To ensure that the reliability of the experiment is intact, will this thesis pursue others work on the same area and try to learn from problems they have encountered. There will also be performed test-experiments, to ensure the correctness and weed out possible problems.

To ensure that the evaluation of the different products does not affect each other, will the

order the products are evaluated be randomized and the evaluation will only be performed with one participant at the time. The participants will then evaluated the products alone and in different order, this to prevent the evaluations being biased. Because of time and resources constraints will this thesis focus on one evaluation alternative, and perform this to full extent. A consequence of this, might be that the evaluation method used does not perform too well. To prevent this will there be conducted an extensive theory study and lessons learned from others work, carefully be taken into account.

4.3 Security testing method

With the problems and difficulties of properly testing security products which incorporates several different protection mechanisms (see Section 3.1), will the security testing method include gathering information from independent security testing laboratories like AV-comparatives.org and AV-test.org. Relevant certifications, like for instance the Virus Bulletin VB100 certification, will also be taken into account.

In testing the security of the different products, or in other words measuring its protection effectiveness, will there be performed a small experiment consisting of testing some chosen elements. A false positive scan will be performed, where a known clean system will be scanned for malware. This scan will also be timed, to check how fast the products are at scanning a computer. The EICAR test file (explained in Section 3.1) will be tested, to make sure that the products works as they suppose to. Steve Gibson's ShieldsUp!, test will be used as a remote port scanning test, to test the most basic firewall capability. Some other firewall leak tests will be performed, to check for outbound protection, process and dll injection and other possible vulnerabilities. A small malware sample set collected from some different sources, will be used to evaluate the detection rates. When handling real malware, will the utmost care be taken to prevent contamination. All tests with malware will also be conducted on a computer not connected to any network.

4.3.1 Reliability and validity

To ensure the reliability and validity of security testing the products, will generally accepted security testing laboratories be a part of the test. With gathering data from their tests, including certificates a product has achieved, will this support or possibly indicate problems with this thesis' security testing experiment. If the results from this thesis are significantly different from other, will this indicate problems with the method.

Because of time and resource limitations, will the malware sample collected only be a small part of the evaluation. To fully evaluate the security effectiveness of the products, would one need a large malware test-set to get significant results. Therefore will other experienced anti-malware testing sources be utilized together in this part.

5 Experimental Work

In this chapter will the experiment and how it was performed be explained.

5.1 Introduction

When addressing the issue of home PC security and usability, is there a lot of considerations one has to take into account. As the usability framework (see Figure 4) explained in Section 3.2.1, one needs certain information when measuring usability. A description about the products together with the desired goals must be explained, and the context of use which includes users, tasks, equipment and environment.

5.2 Products

The software products that will be tested in this experiment, are some Internet Security Suites (see Section 2.3.5 for more information) that aims to provide end-users with a complete security package. These comprehensive security suites, contains most of the security features a end-user needs for protecting their home PC, and put them in an easy-to-use central managed user interface. These products claims to provide the users with full protection against all the traditional threats lurking on the Internet, by giving the users an integrated package of the most common security mechanisms available for personal use.

The following four comprehensive security suites are going to be evaluated:

- F-Secure Internet Security 2008
- Norman Security Suite 2008
- Norton Internet Security 2008
- Trend Micro Internet Security 2008

In Table 5.2, can one see the different main security features these four security suites incorporate (for more information about the products, see Appendix B).

Table 3: The main security features provided by the products.

ID#	Product	Anti-Virus	Anti-Spyware	Personal Firewall	Host-based IDS/IPS
1	F-Secure Internet Security 2008	X	X	X	X
2	Norman Security Suite 2008	X	X	X	
3	Norton Internet Security 2008	X	X	X	X
5	Trend Micro Internet Security 2008	X	X	X	X

See Section 5.5.1, for more description of why these product were chosen.

5.3 Intended objectives

The intended objectives of the thesis is to address the issue of zero effort security for the home PC users. While the intended objectives of the security solutions designed for end-users, is to properly secure their computer. The security suites are designed for being easy-to-use, so that they can be used by almost everybody, regardless of skills and prior knowledge. Some of the security vendors also use the phrase out-of-the-box protection in their advertisement, meaning that potential users do not need to invest much effort into configuring the product.

5.4 Context of use

5.4.1 User

As explained in Section 3.2, (see Figure 3) is there three main dimensions which users' experience differs. Experience with computers in general, experience with the system and knowledge about the domain. In the case of home PC users, one can safely assume that all kinds of different user knowledge and experience are available, from novice users, with very little computer experience, to computer and domain experts.

To accommodate this variety will the experiments incorporate people with both minimal and extensive computer experience, as well as users ignorant and knowledgeable about the domain. The one dimension explained by Nielsen[37] this experiment will not allow for, is the system experience dimension. The fact whether the users are familiar with the system or not, should not affect their ability to participate in the experiment. On the other hand will such knowledge about a system be noted, and displayed in the analysis chapter.

5.4.2 Task

The tasks of the products in questioning, is to secure the computer from the threats lurking on the Internet. To make sure this task is to be fulfilled, incorporates the security suites several different security mechanisms. With the major mechanisms, being anti-virus/anti-spyware, firewall and some also includes intrusion prevention techniques.

These products are often centered around the anti-virus solution, maybe since this is possibly the best publicly known security solutions for personal computers. Another factor to this, might be that these products have evolved from "simple" anti-virus solutions to comprehensive anti-malware solutions incorporating many different security mechanisms into one suite.

5.4.3 Equipment - Generalization of a common home computer

With the variety of different hardware available to the home computer users, are there many considerations to take into account when generalizing a common home computer. The type of equipment, which can vary from stationary computers and laptops too smart phones and PDAs. The type of Operating System (OS) used, most commonly a type of Microsoft Windows, Mac OS or Linux. The performance and capabilities of the hardware, if it for instance is a 32-bit or 64-bit computer. Furthermore is also the type of network equipment like wireless network and Internet connection, important aspects of a common home computer.

Computer performance To describe an average home computer's performance, is there some equipment restriction needed in order too confine the thesis' problem description. Home PC's today vary enormously from household to household, it can be a state of the art stationary computer with multi core CPU, to a several year old computer running on an old operating system. With a quick look on stationary home computers being tested on the Norwegian Internet site DinSide[1], see Table 4, the different capabilities of some common home computers can quickly be illustrated.

Year	CPU	RAM	Hard drive
1999	Intel Pentium III 500MHz	128MB	12,1GB
2003	AMD XP2000+ 1.67GHz	256MB	40GB
2008	Intel Core 2 Duo 3.0GHz	4GB	500GB

Table 4: Three different home PC's tested by Dinside.no, found on their web pages.

Operating System An Internet survey [66] done by XiTi Monitor[2] conducted over the past year (2007), states that approximately 95% of the Internet users uses Microsoft Windows Operating System (see Figure 6). This survey was performed on a perimeter of 152 867 francophone websites audited by XiTi. Out of the 95% of Microsoft Windows users, were over 80% of them using Windows XP.

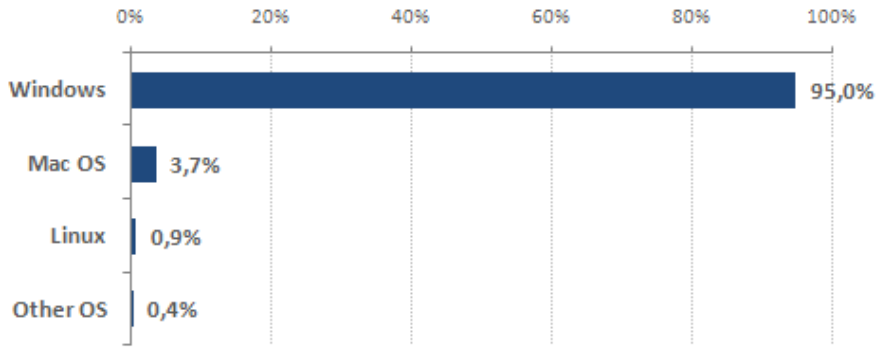

Figure 6: Visit shares of families of operating systems (December 2007), taken from [66].

Internet connection Statistics Norway states in their Internet survey [67] that the total number of broadband connections in private households is by the fourth quarter of 2007 64.2%. The statistics defines broadband as an Internet subscription with a theoretical transmission capacity larger than 128 KBit per second, and thus ruling out any modem and ISDN connection. Key figures gathered from Statistics Norway in 2007, states that 82% of private households has access to a PC and out of these, had 78% access to the Internet.

[1] Dinside.no - is a web page dedicated for the Norwegian consumers and does regularly tests on many consumer items.
[2] XiTi Monitor is a French web survey institute

From the descriptions above, can one argue that a common home computer in Norway has:

- 500MHz-3GHz CPU, 128MB-4GB RAM and at least 10GB-500GB Hard drive.
- Uses Windows XP operating system.
- Has a broadband connection to the Internet.

5.4.4 Environment

With this thesis being about home computer security, is the environment peoples households. Either the computer is used for home office or entertaining purposes, is the environment meant to be personal computer used in private households.

The experiment of course did not have the opportunity to be performed in a private household, but were performed in a restricted access computer laboratory. The participants were also told to act as they would have done in their private household.

5.5 Experiment design

The user experiment were performed on an Intel Pentium 4 1.6GHz with 512MB of RAM and 20GB hard drive. The operating system used, were a fresh and up-to-date Microsoft Windows XP PRO with SP2 install[3]. The reason for the PRO version of Windows XP being used instead of the Home edition, were because this was the available version from Gjøvik University College's MSDN Academic Alliance software center. Another possibility that were considered, was to use virtualization like for instance Microsoft's Virtual PC. This was rejected because one of the products did not support being used in a virtual environment, and the fact that the experiment wanted to simulate a home environment as much as possible. Between each product that was to be evaluated, were the computer reset back to its clean and up-to-date state. This was done with a tool called Paragon Disk Backup, which used approximately five minutes to revert the computer back to its state before the product were installed.

The users, from now called the evaluators, were given some tasks that they needed to perform. The time the evaluator took in perform these task were recorded, and in this part of the experiment did not the observer give any hints or answered questions if not the evaluator were completely stuck. The tasks the evaluators were given, was.

1. Install the product as you would have done at home.
2. Update or make sure that the product is up-to-date, before starting a virus scan of the computer.

After the evaluators had performed these tasks, were they given some time to familiarize themselves with the products, before they started the heuristic evaluation (see Appendix C) of the user interface. In the heuristic evaluation, the evaluators were given a set of heuristics and corresponding statement that they scored on a Likert Scale[4]. The evaluators were told that the statement they scored on the Likert scale, were just an example of the corresponding heuristic and that they were to score the heuristic as much as the statement. As part of the heuristic

[3] During the experiment were service pack 3 released for Windows XP, but this was not installed on the computer.

[4] A Likert scale is a technique for the measurements of attitudes, often represented on a five-point scale from *Strongly disagree* to *Strongly agree*

evaluation, the observer helped the evaluators if they needed it (with for instance explaining the heuristics), and made sure that the evaluation were performed in a reliable and valid manor. The heuristics used were does Nielsen refined in [41]. Some of the heuristics were deliberately left out, since not all the aspects of a heuristic evaluation could be performed within reasonable time for the participators. The aspect of errors, error prevention and help and documentation were eliminated from the heuristic evaluation of the user interface, to enable the evaluators to focus on the other aspects without spending too much time on each product.

After the evaluators were finished with each product were the asked to answer the System Usability Scale (SUS) (see Appendix D), similar to what Hvannberg et. al.[65] did. The SUS scale[49] is a rough measure of how the different users liked the product, and should be answered without thinking and considering too much.

5.5.1 Limitations

To make sure that the thesis did not grasps over too much, were there done some limitations to the experiment. These limitations were also made to ensure that both time and resource constraints will not be a problem for conducting the thesis.

First off, is only security suites available with Norwegian language included into the experiment. This language restriction is incorporated because the population this thesis aims for is the common Norwegian home PC users. This restriction also serves the purpose of limiting the number of security suites that is to be tested. With incorporating to many different security suites will there be a time restriction that can be difficult to meet. Also the fact that the usability part of the experiment consists of actual users who has to perform the evaluation, so is it extremely important to limit the total time needed by them. If the experiment is too long for the participants, will both the willingness to attend the experiment together with their overall performance of the evaluation deteriorate. This problem did Hvannberg et. al. encounter into in [65]. They witnessed the decreased performance of the evaluators because of fatigue. The inspection work for them lasted a few hours per evaluator and therefore become very tedious and tiring. With this in mind, will the usability inspect of the products be restricted too only evaluate certain features of the security suites in questioning and the total duration of the experiment should be kept to a minimum. The evaluators should evaluate the feature he or she wishes, with the restriction that some heuristics are deliberately left out. For instance if the evaluator wants to take a quick peek into the help feature, are they allowed to do so, but the observer would mention that this feature is not taken into account in the evaluation.

5.6 Security testing experiment

The security testing experiment were conducted without using any evaluators, since this was not a direct part of the usability evaluation of the products.

In order to evaluate the different security products identified in Section 5.2, will this experiment consist of several different tasks.

5.6.1 Security features to be tested

Since most of the products are complex solutions that incorporate different security mechanisms, is it important to test and evaluate the different mechanisms. The different features that are to

be tested in the security suites are.

1. The Anti-Malware capabilities
2. The Firewall capabilities

Due to time and resource constrains will other capabilities that the solutions incorporate (see Appendix B for more information) not be tested.

Anti-Malware testing

When testing the anti-malware capabilities of the products will there be performed a false positive test. This will be done by scanning a cleanly installed computer, which is installed from a trusted source. Furthermore will the products be tested with the EICAR test file, to make sure that they work as they suppose to.

The product will also be tested on a relative small malware collection, but with all the problems of anti-malware testing (see Section 3.1.1 for more information) will the anti-malware tests also incorporate results from other independent malware testing facilities.

Firewall testing

Another important security feature of the products, are the firewall. This mechanism prevents unauthorized access to a computer, with checking the incoming packets against a rule set. A firewall also controls the ports of a computer and should make sure that they are not visible from a remote location. Modern firewalls further increases the security of a computer through varies of checks, like for instance application control.

It is therefore very important to security test a firewall, to make sure that it works as it should and keeps your computer safe from hazards. To test the different firewall mechanisms, will there be performed a so called firewall leak test (see Section 3.1.2 for more explanation). These tests checks for instance if the outbound protection (that should be a part of a firewall) works as it should.

Since these security suites contain all their different security mechanisms in one package, can it be difficult to fully assess the different mechanisms. This is important to be aware of, when dealing with such integrated solutions, and hopefully should not be any problem.

6 Analysis and Results

In this chapter will the data and results gathered from the experiment be presented and analyzed.

6.1 Usability experiment

6.1.1 Participants

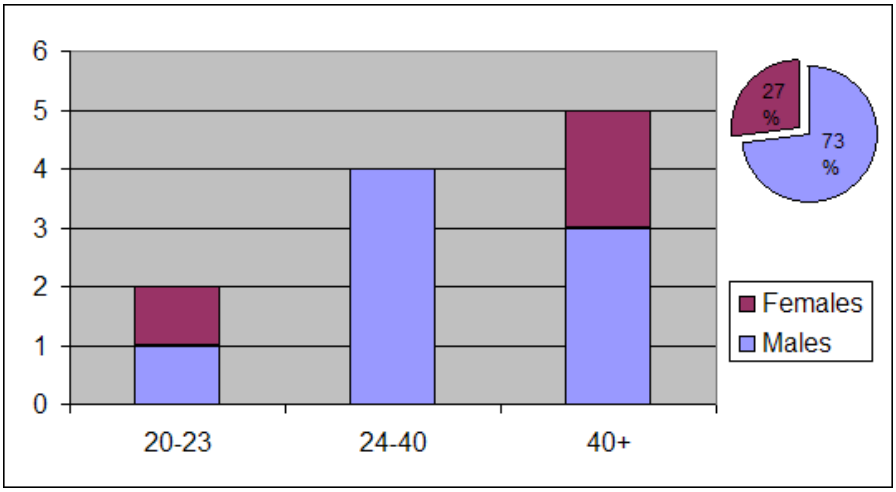

Figure 7: Age and gender distribution of the participants

A total of 11 persons participated in the usability evaluation, 3 females and 8 males, ranging from age 22 to 64 as shown in Figure 7. There were 4 users with minimal computer experience, 2 medium experienced users and 5 experienced computer users. 1 of the participants were a usability expert, and 5 of the participants were information security students. The average total time the participants used, exceeded well over two hours. Some of the participants choose to divide the experiment up in smaller sessions, whilst other wanted to perform it in one session. The ones who performed the experiment in one session, did have the possibility to take breaks if they got tired.

6.1.2 Data gathered from the experiment

In Table 5, one can see the average the evaluators scored on the different heuristic evaluation checkpoints for all products (to see the different heuristic checkpoints see Appendix C, and how the different evaluators individually scored the products see Appendix E). The μ shows the average of all the heuristic checkpoints put together, and can be an indicator on how the evaluators overall rated the user interface of the product in according to the heuristics (the numbers relates to the order of the heuristic and their statements).

1. **Visibility of system status**
   1.1 Each screen of the main menu clearly shows where you are.
   1.2 The interface is understandable and easy to navigate.
2. **Match between system and real world**
   2.1 The language used is familiar and understandable.
3. **User control and freedom**
   3.1 The user is in control of the system and can undo or cancel tasks.
   3.2 The user gets a sense of freedom when using the system (e.g. the ability to move windows around and use the keyboard etc.).
4. **Consistency and standards**
   4.1 Similar information is placed at the approximate same place on each screen.
5. **Recognition rather than recall memory**
   5.1 The system uses seeing and pointing, and provides direct access to the essential tasks.
6. **Flexibility and efficiency of use**
   6.1 The system makes it easy for the user to perform its main tasks.
7. **Aesthetic and minimalist design**
   7.1 The system design is simple and does not include irrelevant information.

	Heuristic evaluation checklist									
Product	**1.1**	**1.2**	**2.1**	**3.1**	**3.2**	**4.1**	**5.1**	**6.1**	**7.1**	μ
1	3.81	3.72	3.27	3.54	3.63	3.72	3.72	3.54	3.72	3.63
2	3.54	3.09	4.09	3.63	2.90	3.63	3.45	3.45	3.63	3.49
3	2.63	2.45	3.45	3.27	3.09	2.81	2.36	2.72	2.81	2.84
4	4.27	4	4.27	3.36	3.45	4.27	3.81	4	3.63	3.89

Table 5: Overall evaluation score for the products.

The average time it took to perform the two tasks the evaluators were given before they evaluated the user interface, can be seen below.

Product 1 (F-Secure Internet Security 2008) The average time it took to install the product for the evaluators were *6:48*, and the evaluators used average *2:08* to update or make sure that the product were up-to-date and starting a virus scan. During the installation answered the evaluators to between 8-12 questions, depending on what choices they made and if they possibly canceled some install configurations.

Product 2 (Norman Security Suite 2008) The average time it took to install the product for the evaluators were *7:06*, and the evaluators used average *2:45* to update or make sure that the product were up-to-date and starting a virus scan. During the installation answered the

evaluators to between 15-19 questions, depending on what choices they made and if they possibly canceled some install configurations.

Product 3 (Norton Internet Security 2008) The average time it took to install the product for the evaluators were *11:57*, and the evaluators used average *6:26* to update or make sure that the product were up-to-date and starting a virus scan. During the installation answered the evaluators to between 4-7 questions, depending on what choices they made and if they possibly canceled some install configurations.

Product 4 (Trend Micro Internet Security 2008) The average time it took to install the product for the evaluators were *5:15*, and the evaluators used average *4:35* to update or make sure that the product were up-to-date and starting a virus scan. During the installation answered the evaluators to between 9-12 questions, depending on what choices they made and if they possibly canceled some install configurations.

The System Usability Scale (see Appendix D) scores, can be seen in Table 6. The System Usability Scale is a "quick and dirty" subjective assessment of usability and calculated to a score between 0-100 (were highest is best usability).

	Participants											
Product	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	μ
1	87.5	85	72.5	60	57.5	100	57.5	40	37.5	40	60	63.4
2	70	92.5	62.5	67.5	42.5	45	57	70	75	35	80	63.4
3	60	45	37.5	10	60	7.5	47.5	70	67.5	0	15	38.2
4	82.5	77.5	67.5	50	95	55	67.5	82.5	80	98	47.5	72.7

Table 6: All the participants System Usability Scale (SUS) scores for the different products.

6.1.3 Data analysis

The evaluators rated the different products based on some provided heuristics and accordingly example statements on a scale from 1-5 (were the highest is the best). This heuristic evaluation of the user interface of the products, can indicate potential usability problems within the product and how comfortable the evaluators were with the interface. A good thing about heuristic evaluations, are that they can possibly indicate were a potential problem within the user interface might be (by referring to the heuristic). For instance can one see from Table 5, that product 3 (Norton) might suffer from some usability issues with their user interface, and especially in heuristic number 5 (recognition rather than recall memory). The different products installation time also varied from an average of just above 5 minutes (product 4, Trend Micro), to almost 12 minutes (product 3, Norton). The differences in installation time varied quite a lot between the evaluators, because some for instance were carefully reading the instructions, while others just pressed next without reading the content. Some participants commented that the installation of some of the products could have for instance been more easy-to-follow, and should show more information about the progress and the remaining time of installation. Some of the inexperienced users, liked the fact that some products gave the user tips and educated a bit on how to secure a computer. The time the participants took to update or make sure that the product

were up-to-date and starting a virus scan, varied also quite a bit. This was because of the fact that some of the products did not easily show the update status, and some of the participants also used quite a lot of time looking for were to update the products and were to start the virus scan. It was not only the novice participants that struggled to update or make sure that some of the products were up-to-date and starting a virus scan, even some of the computer expert had problems with this. The product that had this problem the most, were product 3 (Norton Internet Security). Some of the participants actually needed hints from the observer to perform this task. A reason for this might have been bad language, and the fact that the participants did not have any experience with the product (none of the participants had any experience with the newest versions tested here). Some participants mentioned that, after they had learned the system a bit, was it rather easy to perform this.

The order the different products were evaluated after, were made sure to not be the same for all of the participants. This randomization was done to avoid that the sequence of evaluating the different product did affect the evaluators. If for instance one product were the last to be evaluated for all participants, could this product's evaluation be biased.

The evaluation mainly was performed by non-usability experts, but instead a mixture of novice and expert computer and domain users and the evaluator's subjective notion were captured. This can quite clearly be seen when analyzing the different evaluator's scores (see Appendix E). Some evaluator rates one product very good (giving it high scores all over), while another might rate it very poor. This of course is because the different evaluators emphasizes various things to be important for them, and thus rates the heuristics different. The System Usability Scale (see Table 6) also shows this quite well, with the high variety of scores within the same product.

Some of the potential usability problems the evaluator mentioned in their comments, were:

Product 1 F-Secure Internet security (with Norwegian language) used abbreviations rather much even places it was clearly room for the whole word. Some of these were also not so easy to understand. Many of the participants also mentioned the fact that the entire system felt a bit slow and had too high latency. With the computer used not being a state of the art computer, but yet much more powerful than the minimum requirements of the product, is this a indication that the product is rather resource intensive.

Product 2 Norman Security Suite could have used more dynamic design, with for instance icons which changes accordingly to the use. Some mentioned that the design made it difficult to see the buttons and that some of the submenyes were a bit crooked. The participants also mentioned the fact that the product did not always show the system status or giving an indication that something was about to happen. Some of the more computer and security knowledgeable participants thought the advanced settings were cumbersome to reach.

Product 3 Norton Internet Security used a design that many of the participants did not like. Many of the participants mentioned the fact that the product was not as intuitive as it should have been, and that they needed to remember where things were. This, as mentioned before, is clearly shown in Table 5. This made the learning curve of the product steeper, and some of the participants also felt like they did not get an easy overview of the

product.

Product 4 Trend Micro Internet Security had some issues with the language, some participants also wanted the product to give more detailed information. Some participants mentioned the rather extensive use of dialog-windows as unnecessary. The fact that Trend Micro uses the color red in their logo, and in the icon on the system tray, made it appear to give a notice rather than being neutral. This color also made it difficult to see the warning triangle used on the system tray icon to notify the users about things.

From the overall evaluation score and average (μ), see Table 5, can one quite clearly see that the different products all have some issues with their user interface, and that product 3 (Norton Internet Security) is the "worst" performer. The "best" performer is product 4 (Trend Micro Internet Security), this got the highest score from the evaluators.

6.1.4 Research and findings

The definition by Whitten and Tygar [45] shown below, suggests that usable security software should be based on four basic foundations.

Definition: Security software is usable if the people who are expected to use it:

1. are reliably made aware of the security tasks they need to perform;
2. are able to figure out how to successfully perform those tasks;
3. don't make dangerous errors; and
4. are sufficiently comfortable with the interface to continue using it.

This definition suggests that usable security software should both, have a comfortable user interface for the users to continue using it and that users should (easily) be able to figure out how to successfully perform the relevant security tasks. From the heuristic evaluation, can one argue that the products user interface might suffer from some usability and user-effectiveness related problems. Maybe these problems are related to that the users are not reliably made aware of what security tasks they has to perform, or the fact that these security suites are made to require very little user interaction and thus the user interface is less important. Factors such as economical and social might also be the reason for this.

Performing a heuristic evaluation to test the usability of security suites, can indicate differences in the user interface which might cause problems for the users. According to Whitten and Tygar, should security software have a sufficiently comfortable user interface for the users to continue using it. By performing a heuristic evaluation such as this performed in this thesis helps to make sure that this basic principle is followed. Other usability aspects, such as potential problems and weaknesses, can also be uncovered using a heuristic evaluation. These potential problems can have a very severe effect on security, namely the aspect of users disabling or by other means reduces the security of a system. This problem is very difficult to prevent, some try to solve it by making it hard for the users to actually turn the program off, whilst maybe a better method of solving this is to try to educate the users.

Using John Brooke's System Usability Scale, is another great resource economical method for global assessments of systems usability. Such a questionnaire does not provide as thorough

results as a heuristic evaluation, and is designed to provide a “quick and dirty” usability scale as suppose to the Software Usability Measurement Inventory (SUMI).

6.2 Security testing experiment

This small security testing experiment were conducted in the course of one day, to prevent possible updates and later virus definitions being different in the products. The test on the small malware collection, were conducted in a responsible and careful way in order to prevent spreading and contamination of other systems. To achieve this, were the test machine physically disconnected from any network when real malware were present. When the small malware set were collected, was this done in a virtual machine and with the utmost care.

All security testing were performed with a default installation of the product. The products were also updated with the latest definitions and versions available at the time.

6.2.1 Data gathered from experiment

In Table 7, can one see the number of false positives, see if the product worked properly by detecting the EICAR test file and the time the product took to perform an scan of the entire hard drive (a fresh and up-to-date installation of Windows XP Pro with SP2).

Product	False positives	Detected EICAR	Total time to perform scan
1	0	yes	75:08[1.]
2	0	yes	45:10
3	0	yes	14:17
4	0	yes	18:00

Table 7: False positives and performance

Comments to Table 7:

1. For some strange and unknown reason did the scanner appear to stop several times, this can be why the scan took so long time.

Malware collection detection rates.

In Table 8, can one see how the different product acted on the malware sample and how the detection rate of the product is.

	Products			
Malware categories	F-Secure Internet Security 2008	Norman Security Suite 2008	Norton Internet Security 2008	Trend Micro Internet Security 2008
Backdoor	7/7	6/7[1.]	6/7[1.]	6/7[1.]
Keylogger	3/3	1/3	3/3	3/3
Rootkit	3/3	3/3	3/3	3/3
Trojan	11/11	9/11	9/11	9/11
Virus	4/7	5/7	6/7	7/7
Worm	5/6	6/6	6/6	6/6
Other malware[2.]	0/1	0/1	0/1	1/1
Total	33/38	30/38	33/38	35/38

Table 8: On-demand malware detection rates, showing the number of detected malware samples/out of the total number of malware samples.

Comments to Table 8:

1. One of the Backdoors in the sample set is made for legitimate purposes (legal monitoring of computer activity etc.), but can be used for malicious intentions.
2. The other malware category contains one malware sample which is undefined what category it belongs to.

Leak Tests

In Table 9, can one see how the different product reacted to the leak tests. These test can show how good the firewall capabilities of the products are and can show possible vulnerabilities in the firewall that malware very often takes advantage of. The substitution category is when malicious programs renames itself with the same name of an authorized application. If this fails, the tests succeeds, is this a indication that the security software does not check trusted applications checksum. The launcher category, are programs accessing the Internet via another authorized application, this checks the outbound protection of the firewall. If this fails, is this a indication that the firewall does not check if unknown applications launches other trusted applications. The DLL injection category, checks if the firewall is vulnerable to DLL injections. If this fails, is this a indication for that possible malicious libraries (DLL) can be injected by a parent process. The process injection and the registry injection category are similar to the DLL injection category, indicating that the firewall is susceptible to different injection techniques. The Windows messaging category, are programs attempts to manage the behavior of another through some Windows API messages.

Product	Substitution	Launcher	DLL injection	Process injection	Registry injection	Windows messaging
1[1.]	Failed	Failed	Passed	Failed	Failed	Failed
2	Passed	Failed	Failed	Failed	Failed	Failed
3	Failed	Failed	Failed	Failed	Failed	Failed
4	Failed[2.]	Failed	Passed	Failed	Failed	Failed

Table 9: Leak test results

Comments to Table 9:

1. With the default setting is outbound protection turned off to keep the number of user-interactions down to a minimum.
2. The default setting of choosing home network (not wireless) leads to failing this test, but with choosing a wireless home network (stronger firewall rules activated) is this test passed.

The different leak tests, were all found on Guillaume Kaddouch website[1].

In Steve Gibson's remote port scan, came all products up with all service ports "stealthed". This indicates that the firewall is working as it should and hides all service ports from the Internet.

6.2.2 Data gathered from independent sources

From Andreas Clementi's Anti-Virus Comparative[2], is the latest on-demand detection test of malicious software from February 2008[68]. In this report is the malware collection used for testing the products extremely large, it consist of over 1.6 million samples (1 683 364 too be precise) with the majority being Trojans. The products being tested in this comparative test (16 in total) tests the different vendor's anti-virus engines. The detection rates of the anti-virus engines to the vendor's security suites in questioning will therefore be shown below. As shown in Table 10, is unfortunately Trend Micro not included into this comparatives.

	Products		
Malware categories	F-Secure Anti-Virus 8.0	Norman Security Suite 7.0	Norton Anti-Virus 15.0
Windows viruses	99,7%	94.4%	~100%
Macro viruses	~100%	99.8%	100%
Script viruses	98.7%	75.3%	98.4%
Worms	99.2%	97.1%	99.8%
Backdoors/Bots	97.3%	94.8%	96.0%
Trojan	96.5%	93.2%	97.3%
Other malware	96.9%	77.5%	97.6%
Total	97.5%	94.2%	97.7%

Table 10: AV-comparatives on-demand test from February 2008, see [68] for more information.

[1] `http://www.firewallleaktester.com/index.html`
[2] `http://www.av-comparatives.org/`

AV-Comparatives provides a 3-level-ranking-system (*Standard*, *Advanced* and *Advanced+*). After this comparatives, were F-Secure and Norton rated with Advanced+ and Norman got Advanced.

Andreas Clementi in [69], performs a retrospective test showing the proactive on-demand detection capability that the scanners had in February 2008 with the highest on-demand detection settings over a 1-week test-set. This test is to visualize the anti-virus products ability to detect threats in advance with generic and/or heuristic scanning techniques. Without this ability, are anti-virus products totally reliable on updates to catch new malware which appears every day. This test were performed on the same products as in [68], and the scan engines were updated last on 4th February. For this test, Clementi used new malware samples received between 5th and ~12 to show the proactive detection capabilities that the products had at that time. A note by Clementi is that this test is performed on-demand, and is not an on-execution/behavioral test. This leaves out the ability of some products to be able to detect some samples on-execution or by other monitoring tools, like behavior-blocker, and might give them lower score than in the retrospective test. Clementi also mentions that this specific test, should not be taken as an absolute assessment of quality, but as an indication of who detects more and less. The results of this test, can be seen in Table 11.

	Products		
Proactive detection of malware categories	F-Secure Anti-Virus 8.0	Norman Security Suite 7.0	Norton Anti-Virus 15.0
Number of false positives	very few	many	very few
On-demand scanning speed	slow	average	fast
Windows viruses	58%	71%	6%
Script malware	3%	5%	22%
Worms	9%	25%	24%
Backdoors	4%	45%	23%
Trojans	5%	31%	14%
Other malware	2%	5%	3%
Total	6%	35%	18%

Table 11: AV-comparatives retrospective/proactive test from May 2008, see [69] for more information.

In this test, were a total number of 11 509 malware samples used (for more details see [69]). The certification level the different product achieved were, Norman and Norton reached *Standard* while F-Secure did not reach any certificate in this test.

The ProtectStar TestLab have in [70] made a comparison test of some 2008 Internet security suites. This test have been carried out in cooperation with AV-Comparatives, and the 13 security suites tested includes Trend Micro Internet Security 2008, Symantec's Norton Internet

Security 2008 and F-Secure Internet Security 2008. The malware detection test performed by AV-Comparatives was a on-demand detection test (not a proactive security test) of the malware scanners, similar to that in [68]. The malware test set used where the same as in [68]. In Table 12 can the total malware detection rates from the different security suites relevant for this thesis be seen.

Manufacturer	**Detection in % (Detected samples)**
F-Secure	97.5% (1 641 228)
Symantec	97.7% (1 644 006)
Trend Micro	98.0% (1 649 191)

Table 12: ProtectStar TestLab in cooperation with AV-comparatives on-demand malware detection test performed in February 2008, see [70] for more information.

From the German institute AV-test.org, were an anti-virus comparison test of 30 current anti-malware product performed in March 2008 [71]. In Table 13 can one see how the products relevant to this thesis performed.

	Products			
Tests	**F-Secure**	**Norman**	**Norton**	**Trend Micro**
Detection of malware samples (on-demand, with 1 130 556 samples)	96.8%	92.8%	95.7%	98.7%
Detection of ad- and spyware (on-demand, with 83 054 samples)	93.5%	91.9%	98.6%	95.1%
False positives (scan of 100 000 files)	1 FP	1 FB	0 FB	1 FB
Performance (scan speed)	satisfactory	poor	very good	good
Proactive detection of new, unknown malware	>98%	>95%	>95%	>95%
Response times to new widespread malware	4-6 h	6-8 h	<2 h	2-4 h
Detection of actively running rootkits	>98%	>90%	>98%	>98%
Remediation (cleaning) of malware infections	>95%	>90%	>98%	>95%

Table 13: Current test results of the security suites' malware protection features performed in March 2008, see [71] for more information.

In matousec.com's Firewall Challenge[3] are the firewall capabilities of F-Secure Internet Secu-

[3] See `http://www.matousec.com/projects/firewall-challenge/` for more information

rity 2008, Norton Internet Security 2008 and Trend Micro Internet Security 2008 among other product tested. Out of the three products, did Norton Internet Security score the best with a product score of just 32%, Trend Micro Internet Security scored 27% and F-Secure Internet Security scored a mere 12%.

Certificates

In Table 14, can one see the three known and well respected private-sector certifications anti-malware products can achieve. The latest certificates the products has achieved, are the ones listed in the table. A comment to the table is that, the products does not get these certificates automatically and this might be a reason for why some product has not got it.

Product	**ISCA labs certified**	**Checkmark certification**	**VB100 award**[1.]
1	Yes (Both detection and cleaning certificate)	No	Yes[2.]
2	No	No	Failed, 1 false positive
3	Yes[3.]	No	Yes
4	No[4.]	Yes (Complete Anti-Malware certification)	Failed, 3 wildlist misses, 2 false positives

Table 14: Well known certifications the different products has achieved.

Comments to Table 14:

1. The latest VB100 award, is based on Windows Vista.
2. F-Secure Client Security were tested, which is basically F-Secure Internet Security without parent control.
3. Both Norton AntiVirus 2008 and Norton 360 got detection and cleaning certificate, which utilizes the same anti-malware engine as Norton Internet Security.
4. Trend Micro Internet Security 2007 (32 bit) got both detection and cleaning certificate, but it did not say anything about the 2008 version.

6.2.3 Research and findings

The security tests performed found that there were some differences between the products.

All the products correctly did not have any false positives, and they also detected the EICAR files correct. When it came to the total time it took to perform an entire scan, were there some major differences. Product 3 (Norton) and 4 (Trend Micro) scanned the entire system (20GB) in less than 15 and 20 minutes, whilst product 2 (Norman) used over 45 minutes and product 1 (F-Secure) used 1 hour and 15 minutes.

Data from the leak tests, showed that the firewalls (with default settings) were vulnerable in several different categories, such as outbound protection. Matousec's Firewall Challenge confirms this rather poor performance in their tests as well. One explanation for this can be that, to make

the product as unintrusive as possible for the users, does not the firewall pop-up with a question on what to do if for instance a new outbound connection is made by an unknown program. F-Secure does this with their default settings. The products then rather trusts it capabilities to detect actual malware through other means. Another explanation for the failed leak tests, can be that it is not the firewall that checks for such vulnerabilities but the host-based intrusion prevention technology.

The rather small malware collection that were tested, shows little information on how the real detection rates of the different products are. This because of the problems of getting hold of, sorting and validating an proper sample, is very difficult and resource demanding. This test shows that it some subtle differences between the products, but the sample set is too small to conclude any further.

The independent test by AV-Comparatives, performs a extensive and thorough test on the anti-virus engines. With a malware test set containing over 1.6 million different samples, are the real detection rates shown. Performing such an test, requires many years of expertise and knowledge. The test performed by AV-Test, used over 1.1 million different malware samples, and also had a separate ad- and spyware test which used over 80 thousand different samples. This test also included a false positive scan, performance, proactive detection of new and unknown malware, response time to new widespread malware, detection of actively running rootkits and cleaning of malware infections.

Further data that were gathered from independent source, were the three different and well known certification laboratories. Having these certificates shows that product is valid and performs well, and are carefully used in advertising the products. The fact that product 2 (the Norwegian product Norman) did not have any of these certificates, does not mean that this is a bad product. It almost got the latest (April 2008) VB100 certificate, it only missed with one false positive and the other two certificates are more common in the USA.

6.3 Results

From the usability evaluation performed on these comprehensive security suites, did product 4 (Trend Micro Internet Security) score the best. This does not mean that this is the product available with the best usability. But out of the usability experiment performed in this thesis, with this method and these participants, did this product score overall the best. This product scored the best in the heuristic evaluation as well as in the System Usability Scale. From the security testing experiments (see Table 8), did product 4 (Trend Micro) score the best with the little malware sample this thesis tested. From the firewall leak tests (see Table 9), is it difficult to draw a conclusion, other than the firewall capabilities of the products might be somewhat questionable. The reason for this might be because of the way the product is designed, with requiring as little interaction with the users as possible. This results alone is of very little value, and therefore were data gathered from independent sources as well. From AV-Comparatives on-demand test from February 2008 (see Table 10), scored product 3 (Norton) slightly better than product 1 (F-Secure). In this test was unfortunately product 4 (Trend Micro) not included, but in an other evaluation (in cooperation with ProtectStar TestLab) was product 4 compared against the same test set as the on-demand test from February 2008. From this tests scored product

4, marginally better than the others. From the AV-Test comparison (see Table 13), scored all products quite good in the different tests if one disregards the scan speed test. Depending on what the important features of the test were, did only product 2 (Norman) stand out in a negative way compared to the others.

From the experiments and data gathered from independent sources, can one see that based on this thesis usability experiment and the independent security tests, did product 4 (Trend Micro Internet Security) do well in both security and usability. The empirical data gathered from this thesis (using this method) is that with these products tested, is there differences between the usability and security effectiveness. Even though there exist differences, does it not mean that one have to sacrifice usability to get security effectiveness.

The concepts of security and usability is often mentioned as two conflicting dimensions, although this is true in many cases, are the security and usability practitioners beginning to see that it might be possible to provide both. As today's products getting better (for the most part) from the end-users perspective, and the two previous conflicting dimensions are becoming more important, will hopefully the future bring even more usable and secure software.

Since both usability and security being important for most systems, should it not only be considered early and iteratively in a design and implementation process, but also together. As Yee points out in his article [72], can conflicts between security and usability goals be avoided by considering the goals together throughout an iterative design process. In Garfinkel's PhD thesis [73], is this further explained together with design principles and patterns for simultaneously secure and usable computer systems. Closer collaboration between security and usability not just in making usable security technologies, but also moving to the design of useful secure applications [74] is making the notion of zero effort security for the home PC users a possibility. With proper knowledge and education, together with making software and Operating Systems more secure and usable, will hopefully the common home computer users not having to use too much effort on security.

7 Discussion

In this chapter will the method, experiments and results be discussed. Other aspects of the thesis will also be mentioned.

7.1 Method

To perform the usability evaluation of the products were heuristic evaluation chosen. This form of "discount" user interface evaluation worked quite well. The evaluators, ranking from novice computer users, to experienced computer and domain users and a usability expert, performed the evaluation very satisfactory. There were some signs of fatigue toward the end of the evaluation for some of the evaluators, but with the ability to take breaks and dividing the experiment up into pieces helped tremendously. The System Usability Scale answered right after the product had been evaluated, was also useful. This questionnaire was quick, and provided a good additional information source which verified the overall usability ratings from the heuristic evaluation.

If the evaluation method used had been able to capture the objective opinions of the participants, could possibly the outcome of the evaluation been different. This might be a weakness not only with this method, but with many methods using real end-users.

One issue with the heuristic evaluation, in order for the participants to not spend to much time evaluating the different product, were that some of the heuristics were deliberately left out of the evaluation (the heuristics concerning errors and help and documentation were left out). It could have been interesting to see all Nielsen's refined heuristics been evaluated. But to keep the total time of the experiment not succeed to long over 2 hours per participant, was some aspect omitted. If all Nielsen's 10 heuristics would have been evaluated, would this affected the quality of the evaluation since it would have demanded much time by the participants.

One problem with the heuristic evaluation checklist (see Appendix C), were that it was not so clear for the evaluators what they were supposed to evaluate. A reason for this was because most of the participants were not usability experts and therefore had to have the heuristics (and what they meant) explained for them. The fact that the statement which they scored on the scale from 1-5 were only a part of the heuristic and it was the heuristic that was the important aspect were also a problem, which the observer of the experiment had to explain to the participants. This could have been avoided by making a more clearly heuristic evaluation checklist, or by choosing not use such a checklist at all.

The number of participants performing the usability experiment was rather few, but with this being a heuristic evaluation was there not need for too many evaluators. As mentioned in Section 3.2 (especially see Figure 5), is the number of evaluators needed in a heuristic evaluation much lower than some other usability evaluation methods. With this thesis being interested in deploying resource economical method, was the total number of participants not as important as the method being tested. The fact that the usability experiment consisted of 5 information security students, could have been a couple too many. Since this evaluation mostly captured the

participant's subjective opinion, might the number of computer and domain experts not be as bad after all.

When measuring usability and security effectiveness in security software, is the method used very important. Depending on the software and its application as well as time and resources available, could the method used in this thesis be applicable. With performing a heuristic evaluation, together with the post-test questionnaire System Usability Scale, can one with limited resources roughly measure the usability of the product. Using respected independent sources for gathering the security effectiveness of the products is also a resource economical approach. With other time and resources available, could for instance a more thorough security testing be performed in-house.

7.2 Experiments and results

During the participant's examination of the user interface, were several issues commented on, and subjectively added into the evaluation by the participants. Some evaluators liked the fact that they were educated by the product, while other wanted less "annoying" information that diverted their attention. Some wanted the default settings to be the settings they used, while other wanted more control of what happened without having to manually examine every possible configuration. With these kinds of differences between the users, is it understandable that it is extremely difficult to accommodate an entire population of different opinions in one user interface design. A feature that some evaluators mentioned they would have like to see, were the possibility of switching from a simple to an advanced user interface.

All the participants manage to install the product without any major problems, they also manage to update or make sure that the product was up-to-date with just a little fumbling about. Managing this, ensure that the computer has the protection it needs. The question is then if this protection is enough.

The security experiments showed that these products can provide quite good security to a personal computer, but that they also should be further strengthen with smart and usable security features. For instance can it be very dangerous to rely on too much interaction with the user for a common home computer user, but it can be just as dangerous to provide too little. With malware getting more common and sophisticated, should security suites get equally sophisticated without forgetting to improve their core mechanisms. With the product failing most of the leak test, could be an indication on trade-offs with the design between usability and security.

Problems with data gathered from the independent sources, are that it might be difficult to confirm these and sometimes is there very little information on how they managed to get the results they did. It is therefore important to gather data from respected sources, as well as being critical to this data. In this thesis is the institutes AV-Comparatives and AV-test used a lot for evaluating the security effectiveness of the security suites. Because of the problems with testing anti-malware solutions together with time and resource constrains, was this thesis not able to perform more security testing than it did.

7.3 Security and usability

These security suites provides home computer users with most of the security mechanisms they need to protect their computer, they also get more usable and user-friendly. Such an evolution makes the notion of zero effort security for home PC users possible, maybe not today but possibly in the future. From the problems Johnny encountered in 1999 when he tried, and failed, to encrypt his e-mails. To today's systems that works out-of-the-box, providing the users with all kinds of security features integrated into one package. Can one agree that the usability of security products have improved a lot. This could be because of people have been more comfortable with using computers and the fact that usability have become an important aspect of computers systems, together with the fact that there is more focus on computer security in the general public now than for a decade ago.

With the improving usability of security mechanisms, can hopefully unintended security threats and security related user mistakes become rarer.

7.3.1 Advantages with security suites

The main advantages which security suites offer home PC users are all-in-one security with centralized management. This eases the effort needed by the users to protect their computer. With security suites being more usable and easy to manage for the common home computer user, are they improving the notion of usable (and practical) secure programs.

Security suites can provide additional security compared to specialized software, because it got many security mechanisms integrated which users might not get if they chooses to use individual programs instead.

7.3.2 Disadvantages with security suites

Disadvantage with security suites are that, if they are attacked or turned off is the computer left unprotected. With for instance an individual firewall and anti-virus solutions, will still the firewall protect the computer if the anti-virus crashes. A vulnerability with security suites are that, specially designed compressed files can automatically (and without any user interaction) be downloaded onto the victim's computer. And when either the auto-protect part of the security suites, or an automatically scan later on scans the malicious file might the security suite crash and leave the system unprotected. Some vendors try to reduce this effect by splitting up the mechanisms into different processes, to avoid letting a system become totally unprotected.

Another disadvantage with security suites, are that the user might think that they are properly protected when they are indeed not. As seen in [75] can security software introduce vulnerabilities to a system, articles from the media can also indicate this (see Appendix A). Security software can also become a silver bullet and thus reduces the awareness of the users, making them believe that they are fully protected to do what ever they want without thinking about the consequences. These suites can give the users a false sense of security, by the users putting to much faith into the product.

These comprehensive security suites, also removes the users ability to choice different top-of-the-line security products, and forces the users to rely on the suites efficiency. Performance issues with security suites can also be a disadvantage, since they monitors and scans files automatically to be able to provide proactive security, can they be quite resource demanding. This can lead

to discontent users, which might result in they disabling security features or the entire product altogether.

8 Further Work

Throughout the work on this thesis, have there been identified several other directions that could have been interesting to follow. Due to time and resource constraints will these directions only be mentioned in this section.

Usability testing

To fully evaluate the usability of the products should the user testing be performed over a longer period of time, not just evaluating the product in one (or two) rather short sessions. It would have been ideal to have real users testing the products in their private household over a period of time, for instance a week or month. With such testing, could usability problems that only appear after a period of time be evaluated as well. This would give the total evaluation a more realistic setting, since these products always should be present on the computer. The use of cognitive walkthrough instead of heuristic evaluation, would also have been interesting to see the results of.

Another issue that would have been interesting to pursue, is to have an unaided user experiment, where the users are given a set of tasks to perform. These tasks could be performed within a given time frame and without an observer. This could have shown what tasks the different users managed to perform and not, and give important information about actual usability problems with the products. If one were to perform such an experiment, would it be best to use a quantitative method.

Furthermore could it be interesting to evaluate the usability aspects of some chosen security critical elements, to evaluate which method is best to use in different situations. Such a method can possibly indicate different methods that fit the different user groups, for instance to see how a novice user prefers the setup and design compared to an expert user. This could aid in designing user interfaces which accommodates the different user groups, and such an experiment should maybe also be quantitative.

It would also be interesting to see how home PC users deals with the security of their own computers, to investigate if choices that affect the total security (both in a positive or a negative way) are done because of usability problems, performance issues, lack of knowledge or wrongfully information. Such a direction would have been drastically different than what this thesis became.

Security testing

To test anti-malware products is there a need for a generally accepted objective testing standard and best practices, this is the main objective of the Anti-Malware Testing Standard Organization (AMTSO) to provide. When they develop and releases such a standard, would it be interesting to try it out on the security products.

In the security testing performed, could it have been a good idea to rate the different tests and results on a set point scale. With such an approach would it have been easier to measure

the products up against each other, but using such a method could also possibly give a wrong picture. The product would then have been scored based on the evaluation criteria, which might not have suited the different user groups within home PC users. If such a criteria would have been weighted badly (for instance if some not so important elements was getting to much priority), could the results of the evaluation be misleading.

The use of real malware infected system could really show how effective the security is. Such testing must of course have been performed in a controlled environment. For instance could it be interesting to see how good the products were to detect active Trojan and network worms, and not just see how good the on-demand and retrospective detection rates were.

Other security testing methods that would have been interesting to investigate, is penetration testing and using direct attacks on the products. These attacks could have been tested both from an external (over the Internet) and internal (over a Local Area Network) location. This could have shown how good the products were to resist such kinds of attacks. With using known vulnerabilities to attack the system, would give a good indication on how effective protection the products gives. It could also be interesting to test some of the other security functions that the products incorporate, like for instance testing the phishing detection capabilities.

Matousec's Security Software Testing Suite could also have been interesting to fully test. This testing suite is based on the idea of leak tests, and could maybe have suited the products better than just the leak tests. This testing suite also incorporates some performance testing, which should maybe be a part of the security testing. With a security testing method which also includes performance testing, might yield more complete results on how to assess the effectiveness of the products as well as the usability.

9 Conclusion

Home computer security can be a difficult and challenging task for the common home computer user, often requiring much effort from the people properly trying to protect their PC. These users might therefore need security solutions which require little effort, while still being secure and usable. With more sensitive information accessed via the Internet, should the service providers help their users secure their computers. Some service providers already do this by giving away security software. With presenting security software to the home computer users, is it important that the software has good usability and effective security. For being able to evaluate this, is there a need for resource economical methodology on how to assess both the usability and security effectiveness of the products. Throughout the course of this thesis have a method for evaluating some specific security products with special emphasis on usability and security been tested to see how it performed.

To conclude the work performed in this thesis, must we return to the research questions in Section 1.5).

1. *To which extent is zero effort security for the home PC users possible with some of today's security products?*

Comprehensive security suites, such as those tested in this thesis, can to some extend introduce home PC users with the possibility of zero effort security. This because of the security suites providing all the major security mechanisms needed to protect a computer into one package. With these all-in-one security products requiring rather little effort to use, and also being both quite secure and usable, can zero effort security to some extend be possible. At least do such products help to move toward the notion of zero effort security for the home PC users.

Even though these products can help toward minimizing the effort needed by the users to properly protect their computer, can they alone not provide full protection. Therefore to properly protect a home computer, should one use some security software together with keeping the system and its programs up-to-date. Furthermore should the users be educated and act careful when dealing with computers and the Internet, in order to be able to reach zero effort security.

1.1 *What method is appropriate for evaluating the usability of these products?*

The use of the "discount" method heuristic evaluation to assess the usability of these products, has worked rather well. The use of John Brooke's System Usability Scale, confirmed that the heuristic evaluation captured what it was suppose to capture, and was a good additional information resource. With this usability evaluation using mostly real end-users, has it deviated some from the common way of performing heuristic evaluation (where it is common that a small number of usability experts perform the evaluation). Using heuristic evaluation requires a lower number of evaluators, than for instance with the more common user testing approach. The use

of an observer being able to assist the evaluators, made the evaluation method able to utilize the full potential of the participants. One important aspect of this is that the observer had to be extremely aware of the position he had to avoid making the evaluation biased. The fact that this evaluation used 5 information security students, should not have affected the outcome too much. Because of the method used, together with the fact that it were mostly the evaluator's subjective opinion that were captured, is the background of the participants not so important. The cognitive perception of the evaluators was just as important as their background.

One disadvantage with this method were that the products were evaluated based on a rather short session, and possible long-term problems concerning the usability was not shown. For being able to really assess the usability of these products, should they be tested in a real environment (people's private households) over a longer period of time.

1.2 *What method is appropriate for evaluating the security effectiveness of these products?*

The use of smaller experiments testing out some of the different capabilities to the products, together with gathering information from well experienced independent anti-malware testing sources worked quite well. With the lack of a commonly accepted anti-malware testing standard, (AMTSO.org is working on this) it is rather difficult to properly evaluate the security effectiveness of such all-in-one security suites. The fact that testing anti-malware software is a very difficult and resource consuming task, emphasizes the need for a generally accepted methodology.

Due to resource and time constraints was this thesis not able to perform and validate different anti-malware testing techniques such as those used by for instance AV-Comparatives.org and AV-tests.org. Furthermore could different types of anti-malware testing techniques such as testing other aspects of the products, testing the products against different attacks and drive-by downloading of malware, be interesting to see. This could have given a broader and better picture of the total security effectiveness of these products.

1.3 *What trade-off is there between security effectiveness and usability in today's security products?*

Some of the trade-off between security and usability are shown in the different security tests performed. For instance did the firewall leak tests show that the products had made some adjustment to reduce the effort needed by the users with the default settings. With for instance not enforcing too strict firewall rules, is the user involvement and effort reduced. If the products did not do this, could it have increased the effort needed and reduced the usability of the products. Since pop-up questions often can be very cryptic, and the common home computer users do not know what to do, will the total usability of the products decrease.

To fully investigate the trade-off between the security effectiveness and usability, should the usability experiments been performed over a longer period. This could have possibly shown other trade-offs, which might be present in the products tested.

2. *Which vulnerabilities and threats are of current interest?*

The biggest and most common threat for the common home computer user is malware. This kind of threat is a serious problem effecting computer and Internet security. Human vulnerabilities is also of interest, either it is because of misconfiguration, unintentional or because the security products disrupts the users and therefore is turned off.

Bibliography

[1] Rick Lehtinen, D. R. & Sr., G. G. 2006. *Computer Security Basics*. O'Reilly Media, second edition edition.

[2] Schneier, B. 2005. Attack trends: 2004 and 2005. *Queue*, 3(5), 52–53.

[3] Dourish, P., Grinter, E., de la Flor, J. D., & Joseph, M. 2004. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal Ubiquitous Comput.*, 8(6), 391–401.

[4] Cialdini, R. B. June 2000. *Influence: Science and Practice (4th Edition)*. Allyn & Bacon.

[5] Andreassen, F. L. Are the norwegian internet users ready for the new threats to their information? Master's thesis, Gjøvik University College, 2007.

[6] Breithaupt, J. & Merkow, M. 2005. *Information Security: Principles and Practices*. Pearson Education, Inc., Upper Saddle River, New Jersey 07458.

[7] Anderson, R. 1993. Why cryptosystems fail. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, 215–227, New York, NY, USA. ACM.

[8] Mason, S. 2003. Electronic security is a continuous process. *Computer Fraud & Security*, 2003, Issue 1, 13–15.

[9] Bishop, M. A. 2002. *The Art and Science of Computer Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[10] Gollmann, D. 2005. *Computer Security, 2nd Edition*. John Wiley & Sons, Ltd, New York, NY, USA.

[11] Department of Defense Standard. December 1985. Trusted Computer System Evaluation Criteria (TCSEC). DOD 5200.28-STD.

[12] Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. January-March 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on dependable and secure computing*, 1(1), 11–33.

[13] Alberts, C. J. & Dorofee, A. 2002. *Managing Information Security Risks: The Octave Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[14] Kruger, R. & Eloff, J. H. P. 1997. A common criteria framework for the evaluation of information technology systems security. In *SEC'97: Proceedings of the IFIP TC11 13 international conference on Information Security (SEC '97) on Information security in research and business*, 197–209, London, UK, UK. Chapman & Hall, Ltd.

[15] Cheswick, W. R., Bellovin, S. M., & Rubin, A. D. 2003. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[16] Kurose, J. & Ross, K. 2005. *Computer Networking: A Top-Down Approach Featuring The Internet*. Pearson/Addison Wesley, Boston, third edition edition.

[17] Christodorescu, M., Jha, S., Seshia, S. A., Song, D., & Bryant, R. E. 2005. Semantics-aware malware detection. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 32–46, Washington, DC, USA. IEEE Computer Society.

[18] *Scanners of The Year 2000: Heuristics*, volume pp.225-234 of *Proceedings of the Fifth International Virus Bulletin Conference*, 1999.

[19] Zenkin, D. July 2001. Fighting against the invisible enemy: Methods for detecting an unknown virus. *Computers & Security*, 20, 316–321.

[20] Bennett, C. J. 2001. Cookies, web bugs, webcams and cue cats: Patterns of surveillance on the world wide web. *Ethics and Inf. Tech.*, 3(3), 195–208.

[21] Petrovic, S. 2007. Ids/ips definition and classification. Lecture notes from Intrustion Detection and Prevention.

[22] Scarfone, K. & Mell, P. Guide to intrusion detection and prevention systems (idps). Technical report, National Institute of Standards and Technology (NIST), February 2007. NIST Special Publication 800-94.

[23] Ghosh, A. K. & Schwartzbard, A. 1999. A study in using neural networks for anomaly and misuse detection. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, 12–12, Berkeley, CA, USA. USENIX Association.

[24] Herzog, P. December 2006. Open-soruce security testing methodology manual. OSSTMM 2.2. `http://www.isecom.org/osstmm/` (Visited June 2008).

[25] Computer Security Division. Guideline on network security testing. Technical report, National Institute of Standards and Technology, October 2003. NIST Special Publication 800-42.

[26] Common Criteria. August 2005. Common criteria for information techonology security evaluation. Common Criteria v2.3. `http://www.commoncriteriaportal.org/files/ccfiles/ccpart1v2.3.pdf` (Visited June 2008).

[27] Gordon, S. & Ford, R. 2000. Real world anti-virus product review and evaluations - the current state of affairs. In *NISSC Conference*.

[28] Marx, A. 2000. A guide to anti-malware-software testing. *EICAR*, Best Paper Proceedings, 218–253.

[29] Muttik, I. September 2001. Comparing the comparatives. In *Virus Bulletin Conference*, 45–56.

[30] Wells, J. September 2001. Pragmatic anti-virus testing. In *Virus Bulletin*.

[31] Morgenstern, M. & Brosch, T. September 2007. Malware removal - beyond content and context scanning. In *Virus Bulletin Conference Vienna*.

[32] Marx, A. & Dessmann, F. September 2007. The wildlist is dead, long live the wildlist! In *Virus Bulletin Conference Vienna*.

[33] Marx, A. September 2002. Retrospective testing - how good heuristics really work. In *Virus Bulletin Conference New Orleans*.

[34] Christodorescu, M. & Jha, S. 2004. Testing malware detectors. In *ISSTA '04: Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis*, 34–44, New York, NY, USA. ACM Press.

[35] Karnik, A., Goswami, S., & Guha, R. 2007. Detecting obfuscated viruses using cosine similarity analysis. *ams*, 0, 165–170.

[36] Kaddouch, G. May 2004. Firewall's vulnerabilities, a tool to seek them: Leak tests. `http://www.firewallleaktester.com/docs/leaktest.pdf` (Visited June 2008).

[37] Nielsen, J. 1993. *Usability Engineering*. Morgan Kaufmann Publishers, San Francisco.

[38] Bevan, N. 2001. International standards for hci and usability. *Int. J. Hum.-Comput. Stud.*, 55(4), 533–552.

[39] ISO. March 1998. Ergonomic requirements for office work with visual display terminals (vdts) - part 11: Guidance on usability. Internatinoal Standard ISO 9241-11. ISO 9241-11:1998(E).

[40] Rubin, J. 1994. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., New York, NY, USA.

[41] Nielsen, J. 1994. *Usability Inspection Methods*. Wiley, New York.

[42] Nielsen, J. & Molich, R. 1990. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, 249–256, New York, NY, USA. ACM.

[43] Molich, R. & Nielsen, J. 1990. Improving a human-computer dialogue. *Commun. ACM*, 33(3), 338–348.

[44] Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *Int. J. Man-Mach. Stud.*, 36(5), 741–773.

[45] Whitten, A. & Tygar, J. D. 1999. Why johnny can't encrypt: a usability evaluation of pgp 5.0. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, 14–14, Berkeley, CA, USA. USENIX Association.

[46] Saltzer, J. H. & Schroeder, M. D. September 1975. The protection of information in computer systems. *Proceedings of the IEEE*, 63, 1278–1308.

[47] Constantine, L. 1994. Collaborative usability inspections for software. In *Software Development*, San Francisco.

[48] Bevan, N. & Macleod, M. 1994. Usability measurement in context. *Behaviour and Information Technology*, 13, 132–145.

[49] Brooke, J. 1996. Sus - a quick and dirty usability scale. Digital Equipment Corporation. Reading UK.

[50] McGraw, G. Mar-Apr 2004. Software security. *Security & Privacy, IEEE*, 2(2), 80–83.

[51] Ollman, G. 2004. The phishing guide: Understanding and preventing phishing attacks. Next Generation Secuirty Software Ltd. `http://www.nextgenss.com/papers/NISR-WP-Phishing.pdf` (Visited June 2008).

[52] Ollman, G. 2005. The pharming guide: Understanding and preventing dns-related attacks by phishers. Next Generation Secuirty Software Ltd. `http://www.nextgenss.com/papers/ThePharmingGuide.pdf` (Visited June 2008).

[53] Microsoft Malware Protection Center. June 2007. Microsoft security intelligence report (sir). SIR Volume 3 (January through June 2007). `http://www.microsoft.com/security/portal/sir.aspx` (Visited June 2008).

[54] Microsoft Malware Protection Center. April 2008. Microsoft security intelligence report (sir). SIR Volume 4 (July through December 2007). `http://www.microsoft.com/security/portal/sir.aspx` (Visited June 2008).

[55] Symantec. Internet security threat report: Trends for january-june 07. Technical Report Volume XII, Symantec Security Response, September 2007. `http://www.symantec.com/business/theme.jsp?themeid=threatreport` (Visited June 2008).

[56] Symantec. Global internet security threat report: Trends for july-december 07. Technical Report Volume XIII, Symantec Security Response, April 2008. `http://www.symantec.com/business/theme.jsp?themeid=threatreport` (Visited June 2008).

[57] David W. Baker, Steven M. Christey, W. H. H. & Mann, D. E. September 1999. The development of a common enumeration of vulnerabilities and exposures. In *Second International Workshop on Recent Advances in Intusion Detection*.

[58] Chien, E. & Ször, P. September 2002. Blended attacks exploits, vulnerabilities and buffer-overflow techniques in computer viruses. In *Virus Bulletin Conference*.

[59] Gordon, S. February 2003. Virus and vulnerability classification schemes: Standards and integration. Symantec Security Response White Paper.

[60] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., & Modadugu, N. 2007. The ghost in the browser analysis of web-based malware. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 4–4, Berkeley, CA, USA. USENIX Association.

[61] Provos, N., Mavrommatis, P., Rajab, M. A., & Monrose, F. All Your iFRAMEs Point to Us. Technical report, Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA, February 2008.

[62] Zaytsev, O. 2006. *Rootkits, Spyware/Adware, Keyloggers and Backdoors: Detection and Neutralization*. A-List Publishing.

[63] IBM Internet Security Systems. January 2008. X-force 2007 trend statistics. `http://www.iss.net/documents/literature/x-force_2007_trend_statistics_report.pdf` (Visited June 2008).

[64] Cohen, F. *Computer Viruses*. PhD thesis, University of Southern California's School of Engineering, 1985.

[65] Hvannberg, E. T., Law, E. L.-C., & Lárusdóttir, M. K. March 2007. Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers*, 19, 225–240.

[66] XiTi Monitor. January 2008. Operating systems. Web Survey. `http://www.xitimonitor.com/en-us/internet-users-equipment/operating-systems-december-2007/index-1-2-7-116.html` (Visited June 2008).

[67] Statistics Norway. April 2008. Positive effects of broadband investments. The Internet survey, 4th quarter 2007. `http://www.ssb.no/english/subjects/10/03/inet_en/` (Visited June 2008).

[68] Clementi, A. March 2008. Anti-virus comparative no.17 - on-demand detection of malicious software. Online Anti-Virus comparatives. `http://www.av-comparatives.org/seiten/ergebnisse/report17.pdf` (Visited June 2008).

[69] Clementi, A. May 2008. Anti-virus comparative no.18 - proactive/retrospective test. Online Anti-Virus comparatives. `http://www.av-comparatives.org/seiten/ergebnisse/report18.pdf` (Visited June 2008).

[70] ProtectStar TestLab. March 2008. Protectstar award 2008. Online comparative test. `http://www.protectstar-testlab.org/award/protectstar-iss2008test_eng_web.pdf` (Visited June 2008).

[71] AV-Test. March 2008. Anti-virus comparison test of current anti-malware products. Online. `http://www.av-test.org/index.php?lang=0`.

[72] Yee, K.-P. Sept.-Oct. 2004. Aligning security and usability. *Security & Privacy, IEEE*, 2(5), 48–55.

[73] Garfinkel, S. L. *Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable*. PhD thesis, Massachusetts Institute of Technology (MIT), 2005.

[74] Smetters, D. K. & Grinter, R. E. 2002. Moving from the design of usable security technologies to the design of useful secure applications. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, 82–89, New York, NY, USA. ACM.

[75] Morgenstern, M., Marx, A., & Landesman, M. October 2005. Insecurity in security software. In *Virus Bulletin Conference Dublin*.

A Articles from the media

There are a lot of news that relates to this thesis main issue, about home computers not being sufficiently protected. Some of the articles discovered are the following.

Example articles from the media:
(In Norwegian)

- http://www.idg.no/computerworld/article90836.ece
- http://www.idg.no/computerworld/article52568.ece
- http://www.idg.no/computerworld/article35932.ece

Even security vendors websites can not always be trusted:
http://www.idg.no/computerworld/article91082.ece

Even security products can pose as a risk:
(In English)

F-Secure hit with anti-virus vulnerabilities

(http://www.infoworld.com/article/07/05/30/
F-Secure-hit-with-anti-virus-vulnerabilities_1.html)

Consumer attitudes toward Internet security:

http://www.f-secure.com/f-secure/pressroom/news/fs_news_20080228_01_eng.html

The need for a standard malware competitive comparison:
http://www.theregister.co.uk/2007/08/09/anti_virus_testing/

(All the pages were last visited 19th June 2008)

B Manufacturer Data Sheet

Product F-Secure Internet Security

Version 2008

Pricing 595 NOK for one year subscription (3 PC license).

Functions Anti-Virus
Anti-Spyware
Firewall
Host-based Intrusion Prevention System (HIPS)

Operating System Windows 2000 Workstation, Windows XP Professional/Home/Media Center Edition, Windows Vista (32-bit)

System Requirements Windows XP/2000: Intel Pentium III 600Mhz or higher CPU. 256Mb of RAM. 500MB free HD space (300MB for Anti-Virus only). Windows Vista: CPU capable of running Microsoft Vista 32bit. 512MB of RAM. 500MB free HD space (300MB for Anti-Virus only).

Other E-mail scanner, anti-spam, anti-phishing, rootkit scanner, web-traffic scanning, parental control.

Product Norman Security Suite

Version 7

Pricing 649 NOK for one year subscription (3 PC license).

Functions Anti-Virus
Anti-Spyware
Firewall

Operating System Windows Vista Home Basic/ Home Premium/Business/Ultimate Windows® XP with Service Pack 2 Home/XP Pro/XP Media Center Edition Windows® 2000 with Service Pack 4, Update Rollup 1 Note: 64-bits Windows operating system are not supported.

System Requirements Windows XP/2000: Intel Pentium 450+ MHz CPU. 256MB of RAM. 200MB of available disk space. Windows Vista: 1GHz 32-bit (x86) or 64-bit (x64) CPU. 512MB of RAM. 200MB of available disk space.

Other E-mail scanner, Rootkit detection, Parental Controls, Antipharming, Norman Sandbox

Product Symantec's Norton Internet Security

Version 2008

Pricing 699 NOK for one year subscription (3 PC license).

Functions Anti-Virus
Anti-Spyware
Two-way Firewall
Host-based Intrusion Prevention System (HIPS)

Operating System Windows Vista Home Basic/ Home Premium/Business/Ultimate, Windows XP with Service Pack 2 Home/XP Pro/XP Media Center Edition

System Requirements 300MHz or higher CPU. 256MB of RAM. 350MB of available hard disk space.

Other E-mail scanner, rootkit detection, phising detection, anti-spam and parental control.

Product Trend Micro Internet Security

Version 2008

Pricing 599 NOK for one year subscription (3 PC license).

Functions Anti-Virus
Anti-Spyware
Two-way Firewall
Proactive intrusion blocking

Operating System Windows Vista Home Basic/ Home Premium/Business/Ultimate (32/64 bit), Windows XP Home/Pro with Service Pack 1 or 2 (32 bit), Windows XP Media Center Edition 2004 or 2005 (32 bit), Windows XP Tablet PC Edition 2004 or 2005 (32 bit)

System Requirements Windows XP: Intel Pentium 350MHz or higher CPU. 256MB of RAM. 300MB of disk space available for installation. Windows Vista: Intel Pentium 800MHz or higher CPU. 512MB of RAM for Vista Home Basic and 1GB of RAM for Vista Home Premium, Business or Ultimate. 300MB of disk space available for installation.

Other E-mail scanner, rootkit detection, phising detection, anti-spam and parental control.

Heuristic evaluation checklist

User ID:
Product:

- **Visibility of system status**

Each screen of the main menu clearly shows where you are.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment: _____.

The interface is understandable and easy to navigate.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment: _____.

- **Match between system and the real world**

The language used is familiar and understandable.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment: _____.

- **User control and freedom**

The user is in control of the system and can undo or cancel tasks.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment: _____.

The user gets a sense of freedom when using the system (e.g. the ability to move windows around and use the keyboard etc.).

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment: _____.

- **Consistency and standards**

Similar information is placed at the approximate same place on each screen.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment:_____.

- **Recognition rather than recall memory**

The system uses seeing and pointing, and provides direct access to the essential tasks.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment:_____.

- **Flexibility and efficiency of use**

The system makes it easy for the user to perform its main tasks.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment:_____.

- **Aesthetic and minimalist design**

The system design is simple and does not include irrelevant information.

Strongly disagree		Neither agree nor disagree		Strongly agree
1	2	3	4	5

Comment:_____.

System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

E Heuristic evaluation checklist results

Here are each products individual score the evaluator gave on the heuristic evaluation checklist (see Appendix C for the heuristic evaluation checklist the participants answered on a scale from 1-5, where 1 were Strongly disagree and 5 were Strongly agree).

The heuristics evaluation checklist were as follows:

1. **Visibility of system status**
   - 1.1 Each screen of the main menu clearly shows where you are.
   - 1.2 The interface is understandable and easy to navigate.
2. **Match between system and real world**
   - 2.1 The language used is familiar and understandable.
3. **User control and freedom**
   - 3.1 The user is in control of the system and can undo or cancel tasks.
   - 3.2 The user gets a sense of freedom when using the system (e.g. the ability to move windows around and use the keyboard etc.).
4. **Consistency and standards**
   - 4.1 Similar information is placed at the approximate same place on each screen.
5. **Recognition rather than recall memory**
   - 5.1 The system uses seeing and pointing, and provides direct access to the essential tasks.
6. **Flexibility and efficiency of use**
   - 6.1 The system makes it easy for the user to perform its main tasks.
7. **Aesthetic and minimalist design**
   - 7.1 The system design is simple and does not include irrelevant information.

Table 1 shows how the individual evaluators answered the heuristic evaluation checklist for product number 1 (F-Secure Internet Security).

	Heuristic evaluation checklist								
Participant	**1.1**	**1.2**	**2.1**	**3.1**	**3.2**	**4.1**	**5.1**	**6.1**	**7.1**
1	4	5	4	5	4	4	4	4	4
2	5	3	3	4	5	5	5	5	5
3	4	4	3	4	4	4	4	5	4
4	2	4	3	4	4	4	4	2	4
5	4	3	2	4	3	4	3	4	2
6	5	5	5	4	5	5	5	5	5
7	4	3	2	4	3	2	2	2	2
8	2	2	3	1	2	2	4	1	2
9	4	3	2	2	3	2	2	2	4
10	4	4	4	2	2	4	4	5	4
11	4	5	5	5	5	5	4	4	5

Table 1: Product 1 - Heuristic Evaluation Results

Table 2 shows how the individual evaluators answered the heuristic evaluation checklist for product number 2 (Norman Security Suite).

	Heuristic evaluation checklist								
Participant	**1.1**	**1.2**	**2.1**	**3.1**	**3.2**	**4.1**	**5.1**	**6.1**	**7.1**
1	4	4	5	4	3	4	4	4	5
2	5	4	5	5	4	5	5	5	5
3	4	3	4	4	3	4	4	4	4
4	4	4	5	3	2	2	4	2	4
5	2	3	3	4	2	2	1	3	3
6	4	2	4	4	2	3	3	2	3
7	4	2	2	3	2	4	2	3	4
8	4	4	3	4	4	3	3	2	1
9	1	2	4	2	3	4	4	4	4
10	2	2	5	2	2	4	4	4	2
11	5	4	5	5	5	5	4	5	5

Table 2: Product 2 - Heuristic Evaluation Results

Table 3 shows how the individual evaluators answered the heuristic evaluation checklist for product number 3 (Norman Internet Security).

	Heuristic evaluation checklist								
Participant	**1.1**	**1.2**	**2.1**	**3.1**	**3.2**	**4.1**	**5.1**	**6.1**	**7.1**
1	4	3	4	5	4	4	2	4	3
2	3	2	4	3	4	4	4	4	2
3	2	2	4	3	2	3	2	2	3
4	1	1	2	1	3	3	2	1	2
5	2	3	4	4	5	2	2	3	4
6	2	2	2	4	2	2	1	1	2
7	2	2	4	3	3	2	2	3	4
8	5	4	3	5	4	2	2	4	4
9	3	4	3	4	2	4	3	4	2
10	1	1	4	1	1	2	1	1	2
11	4	3	4	3	4	3	5	3	3

Table 3: Product 3 - Heuristic Evaluation Results

Table 4 shows how the individual evaluators answered the heuristic evaluation checklist for product number 4 (Trend Micro Internet Security).

	Heuristic evaluation checklist								
Participant	**1.1**	**1.2**	**2.1**	**3.1**	**3.2**	**4.1**	**5.1**	**6.1**	**7.1**
1	4	5	4	4	3	4	4	3	4
2	5	5	5	4	5	5	5	5	4
3	4	4	4	4	3	4	3	4	4
4	4	3	5	2	2	4	3	5	3
5	5	5	5	4	4	5	5	5	5
6	4	3	5	3	3	4	3	3	3
7	4	4	4	3	4	4	4	3	2
8	4	4	4	4	4	4	4	4	4
9	4	4	3	3	3	4	4	4	3
10	5	5	5	3	5	5	5	5	5
11	4	2	3	3	2	4	2	3	3

Table 4: Product 4 - Heuristic Evaluation Results