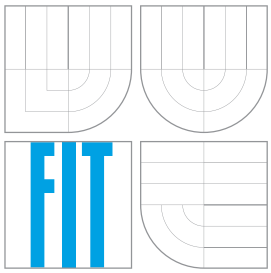
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA BEZPEČNOSTNÍCH INCIDENTŮ V POČÍTAČOVÉ KOMUNIKACI

ANALYSIS OF SECURITY INCIDENTS FROM NETWORK TRAFFIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VILIAM SEREČUN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ONDŘEJ RYŠAVÝ, Ph.D.

BRNO 2015

Abstrakt

Analýza bezpečnostních incidentů se stala velmi důležitým a zajímavým oborem počítačové vědy. Monitorovací nástroje a techniky pomáhají při detekci a prevenci proti tímto škodlivým aktivitám. Tento dokument opisuje počítačové útoky a jejich klasifikaci. Také jsou tady opsaný některé monitorovací nástroje jako Intrusion Detection System nebo NetFlow protokol a jeho monitorovací software. Tento dokument také opisuje konfiguraci experimentální topologie a prezentuje několik experimentů škodlivých aktivit, které byly detailně kontrolovány těmito monitorovacími nástroji.

Abstract

Analysis of network incidents have become a very important and interesting field in Computer Science. Monitoring tools and techniques can help detect and prevent against these malicious activities. This document describes computer attacks and their classification. Several monitoring tools such as Intrusion Detection System or NetFlow protocol and its monitoring software are introduced. It is also described the development of an experimental topology and the results obtained on several experiments involving malicious activity, that were overseen in detail by these monitoring tools.

Klíčová slova

IDS, Snort, NetFlow, NetFlow Analizer, malware, síťové útoky, síťový monitorováni, Trojský kůň, bezpečnostní incidenty.

Keywords

IDS, Snort, NetFlow, NetFlow Analizer, malware, network attacks, network monitoring, Trojan horse, security incidents.

Citace

Viliam Serečun: Analysis of Security Incidents from Network Traffic, bakalářská práce, Brno, FIT VUT v Brně, 2015

Analysis of Security Incidents from Network Traffic

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Ondřeja Ryšavýho Ph.D.

.....
Viliam Serečun
July 28, 2015

Poděkování

I would like to thank Mr. Ing. Ondřej Ryšavý Ph.D. for the supervision, Ms. Maria Luiza Burgarelli Alves dos Santos for the useful advices and grammar corrections and my family for the support. I also would like to thank professor Jean-Marc THIRIET for the useful advices and for allowing me to use part of the project created at IUT Université Grenoble 1.

Contents

Introduction

Security incidents in network traffic have become very common. It is possible to consider as incidents: viruses, DOS attacks or any malicious activities to destroy or access the network without authorization. As a computer network evolves, the number of computer attacks rises. Thus it is indispensable the implementation of security solutions that can provide many possibilities to protect sensitive data of users and organizations.

There are two fundamentally different networks, data networks and synchronous networks comprised of switches. The Internet is considered a data network. Since the current data network consists of computer based routers, information can be obtained by special programs, such as "Trojan horses", planted in the routers [7]. In the synchronous network there is no collecting or buffering of data. For this reason this part of the network is not a target for attackers.

As previously mentioned, the aims of the attackers are mainly personal data or sensitive information. Thus computer and network security is not only a technical issue, but it also includes social engineering. The education of users by network administrators or security experts can make big differences regarding the protection of networks, with measures like setting a password policy [25].

There are several solutions to protect the networks: anti-virus software, firewall, etc. The design and configuration of the network topology also have a big role in security. All these factors can contribute to avoid unauthorized access to sensitive data.

This document will introduce several software solutions to secure network topologies. The classification of computer attacks will help create an idea about the basic attributes and behavior of the attacks. The following text consists of:

- Classification of network attacks
- NetFlow
- Intrusion detection and prevention systems
- Network Topology
- Security incidents

Chapter 1

Classification of network attacks

Fundamentally (as is shown in Figure 1.1), it is possible to divide attacks in two types: active and passive attacks[11]. Every type of attack is specific and requires unique software or methodology for realization. It is possible to consider as a passive attack the scanning of a network topology using various tools (some of them will be described later) or just the monitoring of traffic between two or more devices. Generally it is a preparation for active attacks. Active attacks are more diversified, but also more dangerous. There are many types of active attacks, depending on the target in the network (server, router or client computer) and the goal of the attacker.

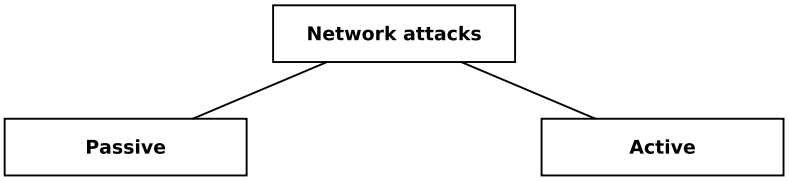


Figure 1.1: Division of network attacks by RFC 2828.

1.1 Passive attack

A passive attack consists of eavesdropping or monitoring the transmission in the network. The goal of the opponent is to obtain the data that is being transmitted. There are two types of passive attacks (as is shown in Figure 1.2): data content monitoring (such as emails, phone calls) and network scanning [22].

Passive attacks are difficult to detect, because they do not involve any unusual activity in the network. Typically, the information is being sent and received through the network in an apparently normal way, and neither sender nor receiver is able to detect that a third party (attacker) is scanning traffic or reading data in the network. However, it is possible to protect the network against these attacks, with firewalls and encryption of the data in the network. Thus, the emphasis when dealing with passive attacks is prevention rather than detection [22].

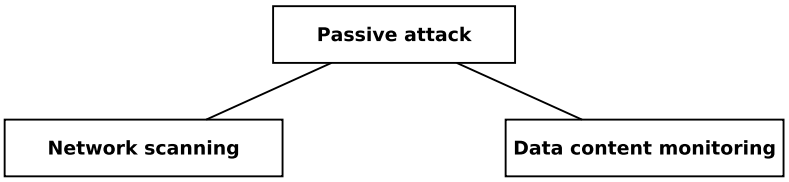


Figure 1.2: Division of passive attacks.

1.1.1 Data content monitoring

Data content monitoring or packet sniffing is an attack, in which the opponent is listening to weak encrypted or non encrypted communication of one or more users and collecting data for future attacks. This data can be used later to access user accounts or to perform a masquerade attack [21]. Figure 1.3 shows an example of packet sniffing scenario.

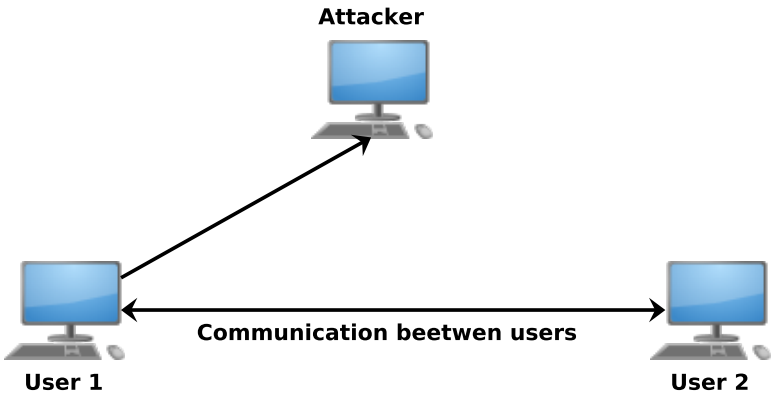


Figure 1.3: Data content monitoring.

Packet sniffing consists of looking at every packet in the network, including packets that are not intended for the receiver. This method analyzes the packets in every TCP/IP layer and provides information for later analysis. Packet sniffers can be run on both non-switched and switched networks [21].

Wireshark

Wireshark[1] is a network protocol analyzer. It is used for analyzing packets and packet sniffing. This network analyzer provides several tools for monitoring and capturing network traffic. Network professionals, security experts, developers, and students around the world use it regularly. In Figure 1.4 is shown the environment provided by Wireshark. It supports most of the computing platforms such as Windows, OS X or Linux [2].

[1]For more information: https://www.wireshark.org.

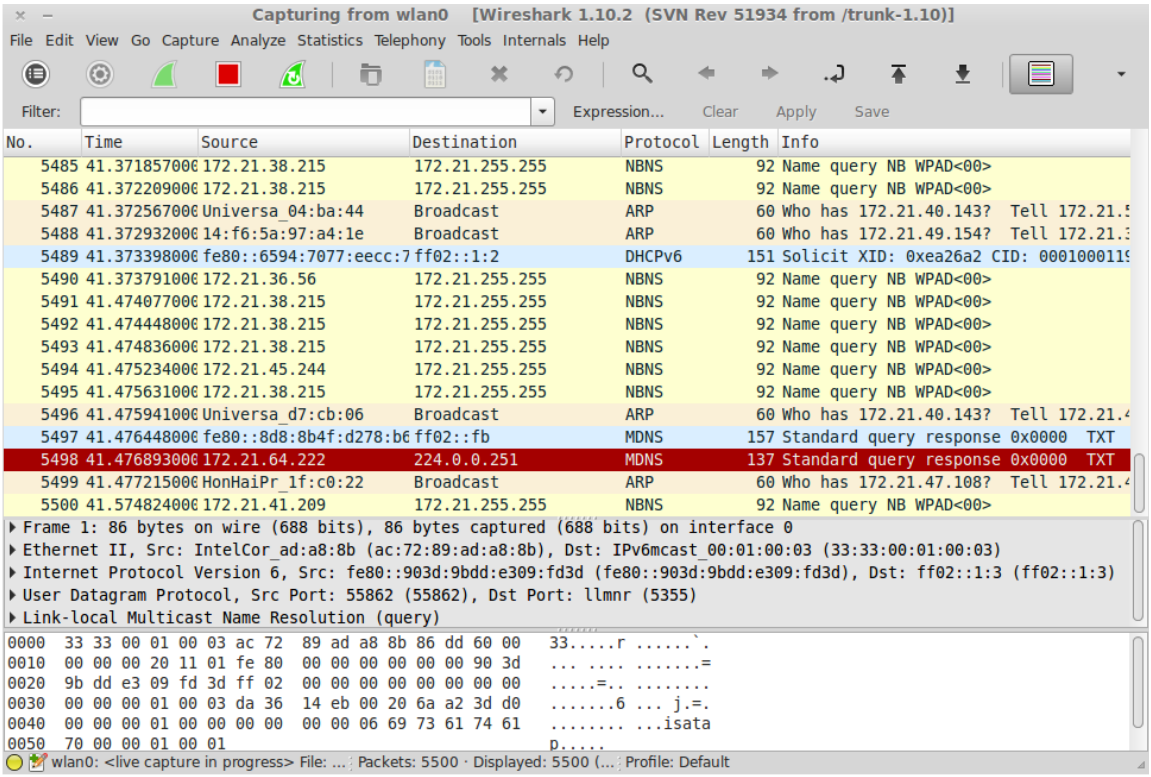


Figure 1.4: Wireshark environment.

1.1.2 Network scanning

Network scanning attack or reconnaissance is a kind of information gathering on the network system and services. Without any knowledge about the network topology, the attacker tries to obtain information from this topology using diversified software and protocols. With this information, the attacker is then able to find weak points in the network and choose a target for the next step. Network scanning can consist of:

- Internet information lookup
- Port Scans
- Ping Sweeps

It is possible to obtain information from the Internet using several tools such as *whois* or *nslookup*. These tools can provide information about the range of IP addresses reserved by an organization, the domain name information from the DNS server or which services are provided by the organization. Afterwards it is possible, by sending packets into the network, to discover which IP addresses are active and how many devices exist in the network. This method is also known as ping sweeping. There is also port scanning, when the attacker tries to discover which ports are open or available. An example of network scanning attack is shown in Figure 1.5 [17].

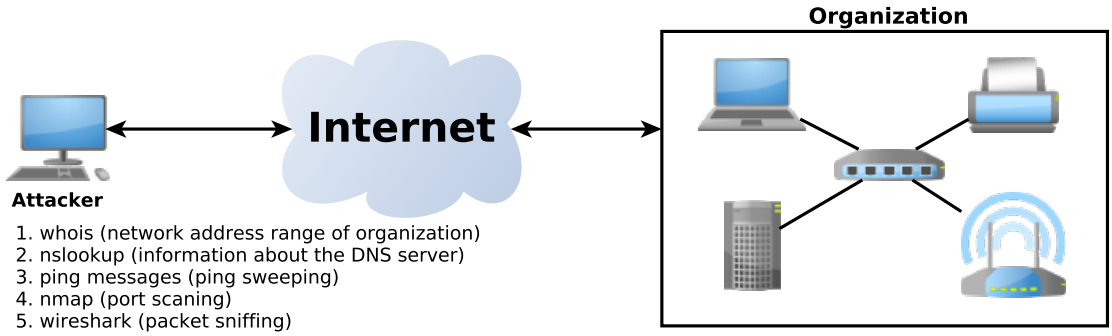


Figure 1.5: Example of reconnaissance attack.

WHOIS

WHOIS is a query response protocol that provides information services to Internet users. This protocol provides information such as registered domain names or the block of IP addresses reserved by an organization, but it can be also used to obtain other information [6].

Ping, FPING

Using ICMP ECHO messages is one of the basic steps to discover active devices in the network. Ping is a tool used to send ICMP ECHO packets and receive ICMP ECHO-REPLAY packets. The use of this tool is acceptable when the goal is to determinate a small number of devices or scan small networks. However, scanning large class A networks can take hours. Thus, it is better to use automatic tools as fping. Unlike the traditional Ping Sweep utilities, that wait for a response from each system before moving on to the next host, FPING is an utility that will send a huge amount of ping packets in parallel, being much faster [4].

Nmap

Nmap is a tool used for port scanning. It sends seven TCP/IP crafted packets and waits for a response. The results are compared with data in the database of known answers. This database is a text file that consists of answers from each known OS. When the answer matches a pattern is possible to guess that the target machine is running in a specific type of OS. Packets are sent to open and closed ports. This tool also provides information about ports and services that are running in the target device [3]. Figure 1.6 shows an example of a Nmap output.

1.2 Active attack

As an active attack is possible to consider every unauthorized intervention in the network, where the attacker modifies data or creates data stream. The attacker in this case is trying to break into or bypass the secure network. Providing protection against this type of attack is more difficult. By the characteristics of the attacks is possible to divide them into three

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-02-01 15:56 CET
Nmap scan report for eva.fit.vutbr.cz (147.229.176.14)
Host is up (0.049s latency).
Not shown: 992 filtered ports
PORT     STATE  SERVICE
22/tcp   open   ssh
80/tcp   open   http
443/tcp  open   https
465/tcp  open   smtps
587/tcp  open   submission
993/tcp  open   imaps
995/tcp  open   pop3s
3128/tcp closed squid-http

Nmap done: 1 IP address (1 host up) scanned in 9.24 seconds
```

Figure 1.6: Example of a Nmap output.

categories, as it is shown in Figure 1.7: masquerade, modification and denial of service [1][11].

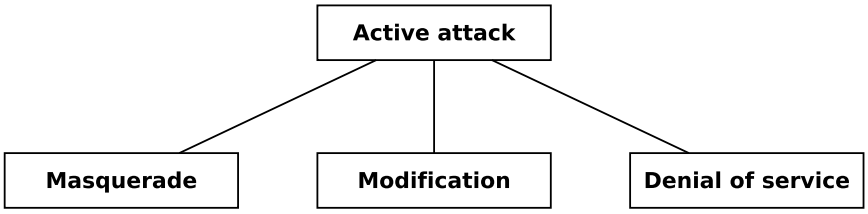


Figure 1.7: Division of active attacks.

1.2.1 Masquerade

The masquerade attack is a kind of attack in which the attacker illegitimately uses the identity of a legitimate user. The aim of this attack is to reach sensitive information such as data or credit cards numbers. A good example is to illegitimately use the identity of a client in a financial transaction [19].

Masquerade attacks can occur in several ways: accessing victim's account and steal sensitive data, or install software for accessing or monitoring such as Rootkit and key logger. As a masquerade attack it is also possible to consider physical access to devices, when the victim leaves his computer logged in and unattended [19].

An example of a masquerade attack is presented in Figure 1.8, where the attacker access the User1 account and with his identity sends messages to User2. It should not be physical or logical access to the victim's computer. It can be also an access to the information system or an email account.

1.2.2 Modification

Modification attack is an attack in which someone in the middle is listening in the communication or modifying this communication, for example the content of the message.

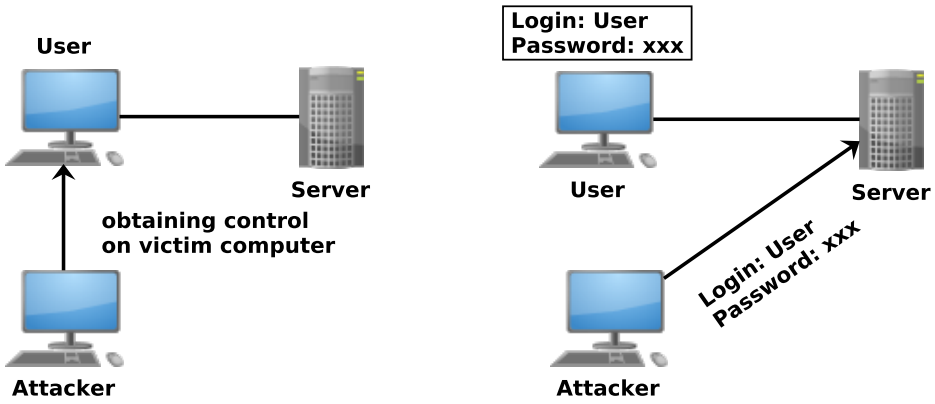


Figure 1.8: Masquerade attacks.

Afterwards, the attacker can send it to the receiving node of the communication or to more users [11].

1.2.3 Denial of service

Danial of service or DoS attack is a malicious attempt where one or more attackers prevent users to access services provided by a device placed in the network, by consuming all the system resources. This type of attack has specific target and most of the time is classified as a very trivial attack although its effectiveness is high. With this attack, the attacker can potentially destroy the network [8][17][22]. An architecture of DoS attack is shown in Figure 1.9 There are several types of DoS attacks but the basic ones are:

- Ping of death
- SYN Flood

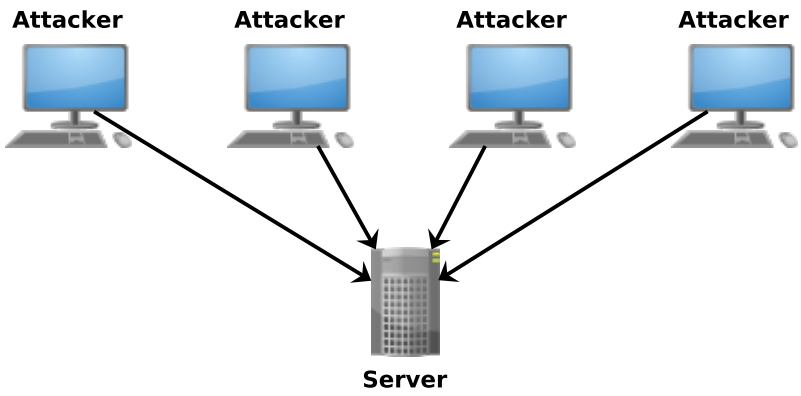


Figure 1.9: Denial of service attack.

Ping of death

The ping of death attack started to be well known in the latest 1990s. This attack takes advantage from old operating systems that are not enough secured. The principle of the attack is to send to a victim's computer ICMP packets, modifying the header of these packets to indicate bigger sizes than the real ones. A normal ICMP packet has 64 or 84 bytes, but after the header modification it can have 65 536 bytes. Sending this size of packet can crash the victim's computer very easily. Recent networks and computers are more resistant to this type of attack [17].

SYN Flood

SYN flood attack uses the three way handshake (Figure 1.10), that is a common method used to establish a connection. The attacker sends a SYN request to the target, then the target replies with an ACK-SYN, but the opponent does not reply with a final ACK. The aim of the SYN flood is to tie up resources in the victim's machine that then will not be able to respond to users [17][12].

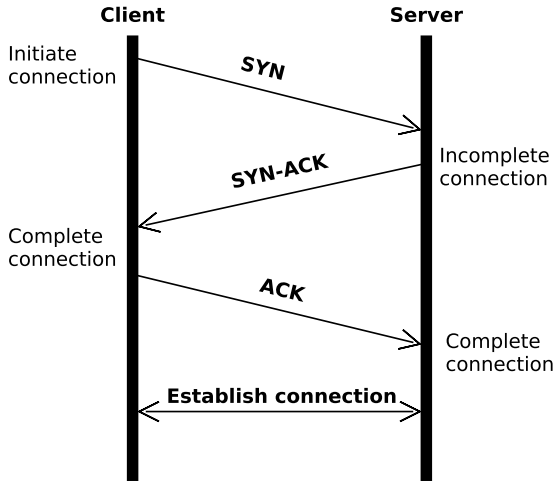


Figure 1.10: Three way handshake.

1.3 Software attacks

It is possible to consider as software attacks any malicious program (software) spread through the network. This software is generally well know as a computer virus, although this is not the perfect definition. An informal definition of "Computer Viruses" was first published in 1984 by Cohen and was followed in 1985 by a formal definition [15].

There are several types of computer viruses, but in this paper it will be introduced only their basic classification and characteristics. The protection against this kind of attack can be provided by anti-virus software. This software is not perfect and can protect users only against well known attacks.

1.3.1 Virus

Computer virus is code or software that replicates and spreads itself. Viruses infect the file system or the operating system and modify the reference to such objects to take control and then multiply again. The existence of a computer virus is easy to recognize when a decrease of the disc space or a slow down of the operating system is detected [22].

1.3.2 Worm

Worm is a malicious program whose aim is to infect the victim's computer and take control of the objects in the system. The difference between viruses and worms is that a computer worm spreads in the network by itself, and does not need the user for that. However, there are also worms that need intervention from the users, such as mailer or mass-mailer worms. The classification of computer worms depends on their characteristics and methods of propagation [15] [24]. There are several types:

- Mailers and Mass-Mailer Worms
- Octopus
- Rabbits

1.3.3 Trojan horse

A Trojan horse is a malicious software that covers its illegal functionality using some helpful software or game. It is very common that this malicious code is part of some open source software since it is very easy to add this illegal functionality in such program. Most of the Trojans create back-doors or loopholes. For example, on UNIX-based systems, attackers add a cover functionality to the *ps* utility (UNIX tool for showing running processes) that can relate to another process, creating a back-door in the victim's system. This type of Trojans are well known as user mode rootkits. Generally, Trojans can also relate to another malicious activity, such as monitoring the victim's system [15][17].

1.4 Other types of attack

In this part will be described and shown some computer attacks using several methods. To prevent these kinds of attack it is important to understand their principles and to be prepared to apply security policies against them.

1.4.1 Distributed denial of service

Distributed denial of service or DDoS is an attack in which several machines infected by the attacker's malicious software simultaneously attack the victim's device or system. There are mainly two types of DDoS attack: classical DDoS attack and distributed reflector DoS (DRDoS) [18].

The classical distributed DoS consists of master zombies and slave zombies. All these devices are infected by the malicious software. The attacker can coordinate only master zombies and the master zombies can coordinate slave zombies. This pyramid structure works in such a way that all devices are in *sleeping mode*, and waiting for an impulse from the attacker. The attacker sends a message with the target specification to the master

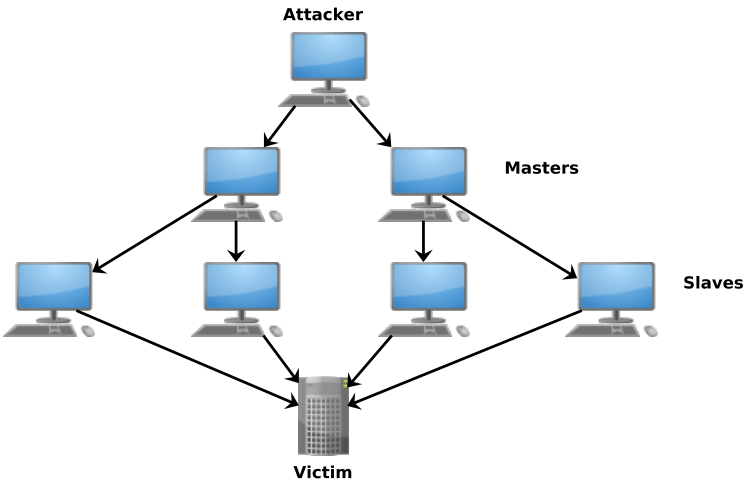


Figure 1.11: Distributed denial of service.

zombie and this master zombie distributes this message to the slave devices that then wake up and start the attack to the victim. The architecture of the classic DDoS attack is shown in Figure 1.11 [18].

The Distributed reflector attack is shown in Figure 1.12 and it is similar to the classical distributed DoS. The difference between them is that in this case the pyramid structure described previously is increased by one layer of the reflector devices. The principle of the attack is the same, although the slave zombies are not attacking victim computers. They send modification messages, for example ICMP messages, where the source address is changed to be the victim's one. This provides to the attacker more devices to attack, and thus it is more detrimental than the classical DDoS [18].

1.4.2 Phishing attack

Phishing is an attack whose aim is to gain sensitive data from the user (such as credit card numbers) or to access information of the user's account. In this case the attackers create a website, for example a fake bank website, that looks like the same as the original one. Using this website they can simulate some catastrophic scenario in which something happened in the bank. This information is then sent to the victim, most of the time by email, informing the user that this catastrophic scenario happened to the organization and they need the user to verify his access information [11].

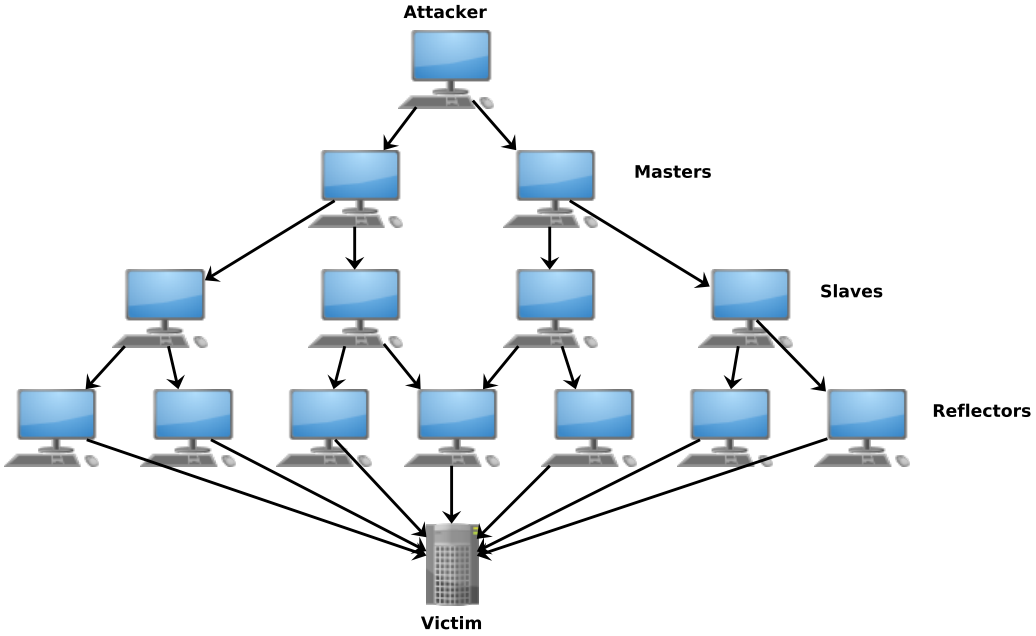


Figure 1.12: Distributed reflector denial of service.

Chapter 2

NetFlow

NetFlow is a monitoring service introduced by Cisco that provides to administrators information about the data network, such as the source and the destination address between communicating nodes, the number of transmitted packets, the size of transmitted packets, etc. (full specification can be found in [5]). This information is grouped into a record called flow or NetFlow and sent from a network element (router, switch) to a station called collector. This station collects the NetFlow data of one or several devices and stores them into a database, making them visible to the user [5].

2.1 Architecture and terminology

The NetFlow service, when implemented in a real environment, consists of several parts. In this section will be described the basic terms of the architecture and the basic terminology commonly used with this technology.

2.1.1 Exporter

An Exporter is a device, such as a router or a switch, that provides the NetFlow service. It means that this device monitors the data from an **observation point** (one or a set of interfaces), and creates flows. These flows are packets passing through the observation point in a certain period of time. The information flow is transferred to a Flow Record and sent to a device called NetFlow Collector [5].

2.1.2 NetFlow Collector

A NetFlow Collector receives Flow Records from one or several Exporters. These packets from the Exporters are processed by this device, parsed and stored as a Flow record information. Practically, the NetFlow Collector receives two types of packets. The first type is the Template Record and the second one is the Flow Record. Template Records generally increase the flexibility of process of the Flow Records, because they provide to the Collector the process of the Flow Record packets without knowing all the interpretation and data in the Flow Record [5].

2.1.3 Others terms

„A **Flow Record** provides information about IP Flow observed in observation point or device interface“ [5].

„A **Template Record** defines information and interpretation of fields in Flow Data Record“ [5].

„A **Flow Data Record** is a data record that contains values of the Flow parameters corresponding to a Template Record“ [5].

„An **IP Flow** or **Flow** is a set of IP packets passing an Observation Point during a time interval. All packets that belong to a particular Flow have a set of common properties (data contained in the packet)“ [5].

„An **Observation Point** is a location in the network where IP packets can be observed (interface/interfaces of a router)“ [5].

2.2 Visualization software

There are several tools available for visualization of the NetFlow data. It processes received packets and stores them in a hard-disk or database where the data can be analyzed or visualized by a visualization software. All visualizations are done by third party companies, since Cisco does not provide this feature. Some of the visualization applications provide complex analysis environment, including the NetFlow Collector as well.

2.2.1 NetFlow Analyzer 11

NetFlow Analyzer 11 is an application that provides collection of NetFlow packets ,or other flow packets (sFlow, IPFIX, etc.), supported by enterprise devices (routers, switches). It then generates traffic reports for better understanding of what is happening in the network. The report consists of several charts providing information about bandwidth, types of packets and many other useful information. This application was created by ManageEngine corporation. It also provides some additional features, such as faster troubleshooting of the network, validation of QoS policies, scheduling of reports, IP grouping, etc. Figure 2.1 shows an example of the graphical environment of the application [14].

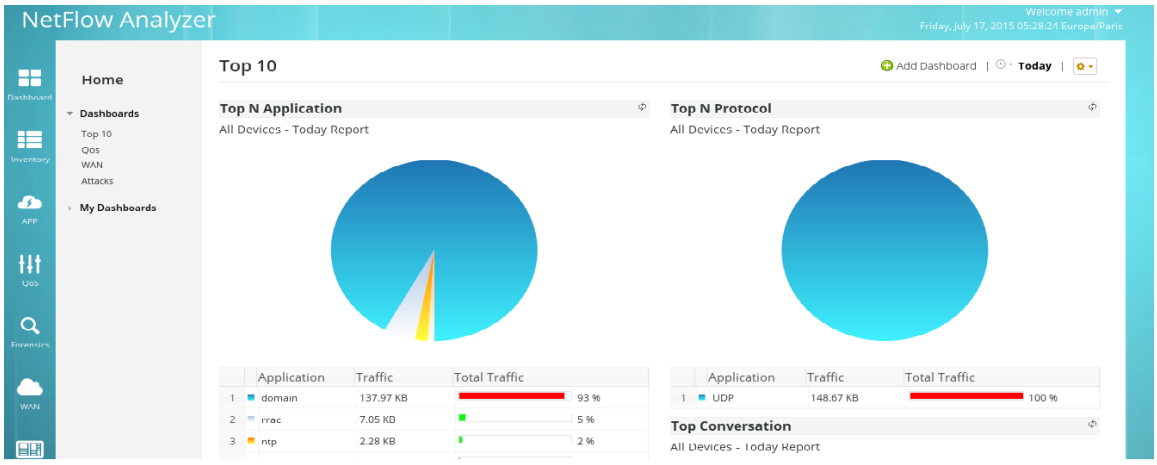


Figure 2.1: Graphical environment of the NetFlow Analyzer 11

Chapter 3

Intrusion detection and prevention systems

Intrusion detection and prevention systems are commonly used in modern network architectures. There are several methods to implement this technology in a network infrastructure. This chapter will describe the basic terminology and architecture of these systems.

3.1 Intrusion detection systems

Intrusion detection is a process of monitoring that analyzes events from computers and network traffic to identify a network intruder or attacker. This automation process is provided by an intrusion detection system. This software mainly provides the following features [20][10]:

- Monitoring and analysis of user and system activity
- Audit of system configurations and vulnerabilities
- Assess of critical system integrity and data files
- Statistical analysis of activity patterns based on matching known attacks
- Abnormal activity analysis
- Operating system audit

3.1.1 Host intrusion detection system

Host intrusion detection system (HIDS) is a software agent in the host system that analyzes data or traffic. Modern hosts run in background on critical machines, such as DNS, MAIL and other types of servers. The HIDS takes a snapshot of your current system and compares it with a previous snapshot. If it detects some modification in the critical files, an alert is sent to the administrator [10][20].

3.1.2 Network intrusion detection system

Network intrusion detection system (NIDS) is an independent platform that collects and analyzes data from network traffic. NIDSs collect the traffic by connecting to different parts

of the network, such as hubs, switches, routers and other network devices. When an attack is identified, the NIDS sends an alert to the network administrator [10][20].

3.1.3 Network node intrusion detection system

Network node intrusion detection system (NNIDS) is very similar to the NIDS. It also analyzes the network traffic. The difference between NIDS and NNIDS is that in the NNIDS the network traffic is monitored only in single hosts [10].

3.1.4 Intrusion detection system components

There are four typical components in network detection systems: sensor, management server, database server and console.

- **Sensors** or agents are used to monitor the network. They are placed in strategic places in the network topology, where they can collect data and send it to the management server. Sensors or agents can be a software or a hardware device. The term sensor is used more frequently in network monitoring systems and the term agent in host-based systems [20].
- **Management server** is a centralized device that collects data from sensors and agents and manages them. Some management servers perform analysis that the agents and sensors are not able to do. This device compares events, analyzes them and matches events from multiple sensors [20].
- **Database** is a component in the architecture where all the alerts from the management server are stored. It can be any kind of database, depending on which type of IDS is supported [20].
- **Console** is a program that provides an interface for the IDS or a configuration interface for sensors and agents. These consoles are installed on standard computers and laptops [20].

3.2 Placement of the IDS in the network

There are several methods to place an intrusion detection system in the network. Every method has its advantages and disadvantages that will be described below. In every scenario, as shown in the example of Figure 3.1, will be used a network topology that consists of two servers, one computer and an IDS station. Red lines will represent the connection between the IDS station and the network traffic. Dashed lines will represent the management connection that is used for accessing or upgrading the station [16].

3.2.1 IDS using hub or spanning port

The scenario in Figure 3.1 shows an IDS station placed under a device, that is redistributing the traffic among the station. This device can be a switch or hub. In the case of using a switch, all traffic is captured and analyzed in a specific port that is configured as a spanning port. Using a hub can cause security risks since all connected devices can read the traffic. The system in the internal network is not at the mercy of an IDS network failure, that could bring the network down. In this type of connection is common to place sensors in the internal network [16].

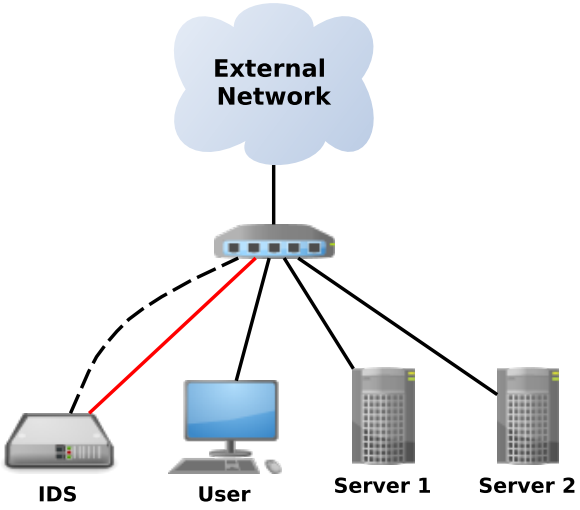


Figure 3.1: IDS using hub or spanning port

3.2.2 IDS using tap

It is possible to connect the IDS to the network tap behind the switch as is shown in Figure 3.2. The network tap ID hardware device transfers information from the network traffic without disrupting it. This concept is not very common, but is useful when it is necessary to setup a host monitoring station or troubleshoot a problem temporarily by the IDS. Also it can be used if there is no switch or hub in the network or the network architecture does not support inline connection. The advantage is that the IDS can analyze all the traffic passing from the internal network to outside [16].

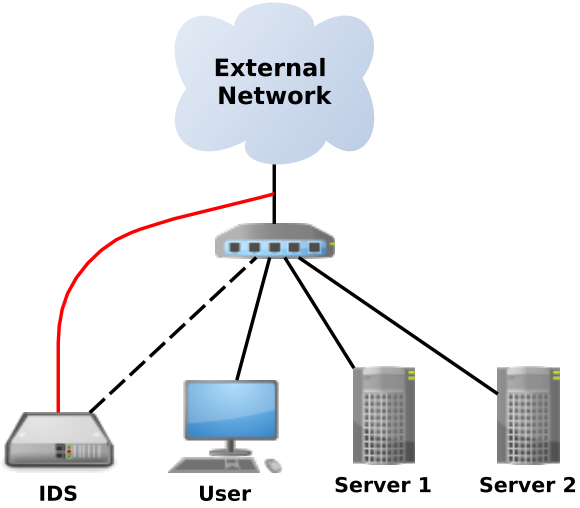


Figure 3.2: IDS using network tap

3.2.3 IDS connected inline

The inline connection of the IDS, another possibility when creating an IDS network infrastructure, is shown in Figure 3.3. The connections represented by red color show the interfaces where the traffic is monitored by the IDS and all the traffic that is passing through these interfaces. The dashed line represents the connection for device management. This architecture is not the best alternative of how to use IDS. „System failure of the IDS failure will prevent systems on the internal network from communicating with external systems“[16]. The advantage of this architecture is that all outcoming and incoming packets will be seen by the IDS. In this scenario is not possible to monitor by the IDS the communication between the devices inside the internal network [16].

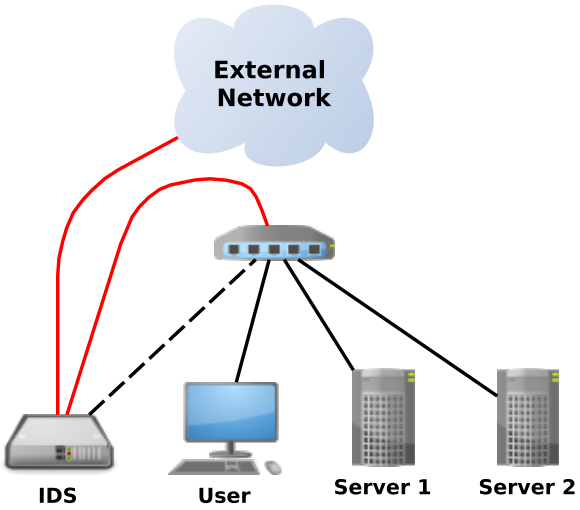


Figure 3.3: IDS connected inline

3.3 Intrusion prevention system

Intrusion prevention system (IPS) is an extension of the intrusion detection system. However, instead of detection it provides prevention before the attack. It is possible to classify it as a firewall in the network. IPSs were created to stop attacks in real time and protect the network. It is an active in-line device that can drop attacks and perform activities to stop attacks, such as disconnect ports before the intrusion happens[9].

Chapter 4

Network topology

This chapter describes the network topology and its configuration. This topology will be used to run several experiments where the main goal is to monitor the behavior of computer attacks and malware.

4.1 Network topology

All network topology is running in a virtual environment (in this case it was used VMWARE Workstation 11). The basic network topology consists of a router, a server running an IDS and a computer connected to a virtual switch. Due to the impossibility of configuring a spanning port in the virtual switch provided by the virtualization software, the IDS was connected inline as described in section 3.2.3. The network topology is shown in Figure 4.1.

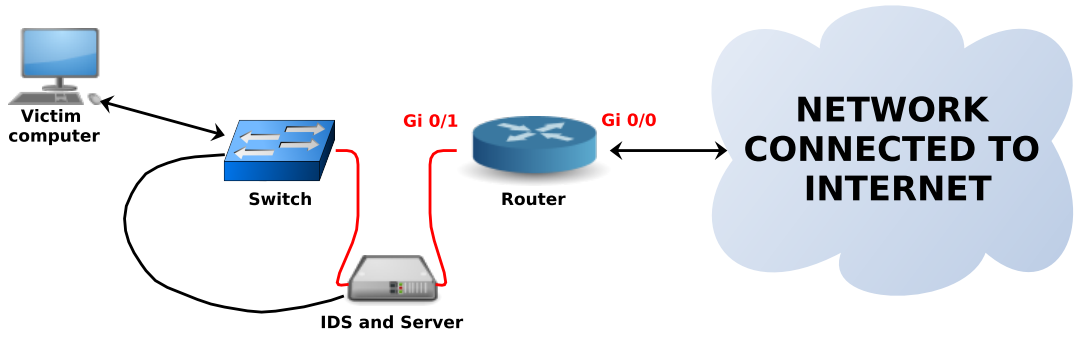


Figure 4.1: Topology used for the experiments

4.2 Router configuration

The router used was a Mikrotik RouterOS[1] for PC. This router was created virtually with two interfaces. One port is used for connection to the INTERNET, and another is connected to the network 192.168.12.0/24. Since the router is placed in a virtual environment, the interface connected to INTERNET is in a NAT network.

[1]For more information: http://www.mikrotik.com.

4.2.1 Interfaces configuration

The router has two interfaces. The first one is connected to the network 192.168.30.0/24 using the IP address 192.168.123.254/24. This network is used as a connection between the router and the IDS station. The second interface is connected to a NAT network 192.168.123.12.0/24 with IP address 192.168.12.50/24. An example of how to add and configure the addresses is shown in Figure 4.2.

```
[admin@Mikrotik] > ip address
[admin@Mikrotik] > add address=192.168.123.50/24 interface=ethernet2
```

Figure 4.2: Example of interface configuration

4.2.2 Routing and NAT

To communicate with the outside network is necessary to configure routing and NAT translation.

For the routing it is necessary to configure a default gateway (in this environment it is 192.168.12.2/24). The static routing also needs to be set to allow the communication between the outside and the internal network (192.168.123.0/24). This routing consists of a rule, in which all incoming traffic is sent to the address 192.168.123.253. This IP address represents the network interface of the IDS. The traffic from this interface is redirected to the internal network. An example of the routing configuration is shown in Figure 4.3.

```
[admin@Mikrotik] > ip route
[admin@Mikrotik] > add gateway=192.168.12.2
[admin@Mikrotik] > add dst-address=192.168.123.0/24 pref-src=192.168.30.254
gateway=192.168.30.253
```

Figure 4.3: Example of the routing configuration

The NAT translation is set in the router interface (IP address 192.168.30.254/24). All the traffic from the networks 192.168.123.0/24 and 192.168.30.0/24 is translated to the address 192.168.12.50. Figure 4.4 shows an example of NAT configuration in a Mikrotik router.

```
[admin@Mikrotik] > ip firewall nat
[admin@Mikrotik] > add chain=srcnat action=masquerade
to-addresses=192.168.12.50 src-address=192.168.30.0/24
[admin@Mikrotik] > add chain=srcnat action=masquerade
to-addresses=192.168.12.50 src-address=192.168.123.0/24
```

Figure 4.4: Example of NAT configuration

4.2.3 NTP client

The NTP client was configured to allow time synchronization of all devices in the network. The NTP server is running on the address 192.168.123.1/24. Figure 4.5 shows an example of the NTP client configuration.

```
[admin@Mikrotik] > system ntp client
[admin@Mikrotik] > set primary-ntp=192.168.123.1 mode=unicast enabled=yes
```

Figure 4.5: Example of NTP client configuration

4.2.4 TrafficFlow

To monitor and analyze the traffic passing through the router it was configured a NetFlow service. This service is represented in the Mikrotik router as a TrafficFlow service. The Exporter is configured to send data to the NetFlow Collector (IP address 192.168.123.1/24) in version 9. The configuration of the TrafficFlow is shown in Figure 4.6.

```
[admin@Mikrotik] > ip traffic-flow target
[admin@Mikrotik] > add address=192.168.123.1:9996 version=9
[admin@Mikrotik] > enable numbers=0
```

Figure 4.6: Example of TrafficFlow configuration

4.3 Server configuration

For experiments it was used CentOS 7 server. This server provides several services (DNS, DHCP, etc.) and was also used as an IDS monitoring station connected inline. This server has 3 interfaces, where the first one is used for service distribution. The second interface is used as a default gateway with all the traffic being redirected to a third port connected to a different network, which is used as a connection between the router and the IDS monitoring station. This connection makes it possible to monitor all in-coming and out-coming traffic.

Before any configuration is done it is necessary to disable SELinux. The configuration file is placed in */etc/selinux/config*. In this file it is required to set the option **SELINUX=disable**. After that it is necessary to restart the server.

For better monitoring of the network it is recommended to disable the firewall. An example of how to disable the firewall on CentOS 7 is shown in Figure 4.7.

```
[root@zeus ~]# service firewalld stop
[root@zeus ~]# chkconfig firewalld off
```

Figure 4.7: Example of how to disable firewalld

4.3.1 Interface configuration

As a first step to configure the interfaces it is recommended to turn off the NetworkManager service. It is possible to disable this service using two commands, as shown in Figure 4.8.

```
[root@zeus ~]# service NetworkManager stop
[root@zeus ~]# chkconfig NetworkManager off
```

Figure 4.8: Example of how to disable NetworkManager

The files to configure the interfaces are placed in */etc/sysconfig/network-scripts*. In this directory are placed all the configuration scripts of the network interfaces and the routing rules. The configuration files start with the prefix **ifcfg-***interface_name*. In these configuration files was necessary to add or change several options. These options manage static IP address configuration of the interface, interface start when the system boots, DNS, etc. The default gateway needs to be set only in the interface belonging to the IDS (192.168.33.0/24). In the interface that provides services for the internal network is necessary to set the DNS option to a local address. The option NM_CONTROLLED manages if the interface is configured by the NetworkManager service. All the options are similar as is shown in Figure 4.9, but with different IP address, default gateway and DNS.

```
...
BOOTPROTO=no
ONBOOT=yes
IPADDR=192.168.123.253
NETMASK=255.255.255.0
GATEWAY=192.168.33.254
NM_CONTROLLED=no
...
```

Figure 4.9: Part of the interface configuration file

4.3.2 DNS

The installation of the DNS required two packages: **bind** and **bind-utils**. In the main configuration file (placed in */etv/named.conf*) is necessary to set the forwarders and add two records, specifying a new zone. The internal network is placed in the *leopard.org* domain. The zone files are placed in the */var/named* directory. There are two files, a zone file *leopard.org* and a reverse zone file *123.168.192.db*. An example of the options required for setting the DNS in *named.conf* is shown in Figure 4.10. The DNS zone files are shown in appendix B.

4.3.3 DHCP

The DHCP service required the installation of the **dhcp** package. The configuration is located in */etc/dhcp/dhcpd.conf* file. In this file is necessary to set several options. The range of IP addresses provided by the server in the testing topology goes from 192.168.123.50 to 192.168.123.100. There is also the possibility of setting IP addresses for the router, SNTP server, etc. This service was included to test the topology to automatic obtain IP addresses

```
options {
       ...
       listen-on port 53 {127.0.0.1; 192.168.123.1};
       allows-query {any};
       forwarders {
            130.190.190.4;
            8.8.8.8;
            8.8.4.4;
       };
};

...

zone "leopard.org" IN {
       type master;
       file "leoprd.org";
       allow-update {none;};
};

zone "123.168.192.in-addr.arpa" IN {
       type master;
       file "123.168.192.db";
       allow-update {none;};
};
```

Figure 4.10: Example of *named.conf* file

for the computers connected in the internal network. Figure 4.11 shows the configuration file of the DHCP service.

```
option domain-name      "leopard.org";
option domain-name-servers      192.168.123.1;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.123.0 netmask 255.255.255.0 {
       range 192.168.123.50 192.168.123.100;
       option broadcast-address 192.168.123.255;
       option routers 192.168.123.254;
       option subnet-mask 255.255.255.0;
       option ntp-servers 192.168.123.1;
}
```

Figure 4.11: DHCP configuration file

To make DHCP listens only on specific ports it is necessary to copy the file from */usr/lib/systemd/dhcpd.service* to the directory placed in */etc/systemd/system/*. In this file it is required to add the name of the network interface (for ex. eno33554984) in the end of the line starting with **ExecStart=**. After this modification it is necessary to reload the configuration with **"$ systemctl --system deamon-reload"** and restart the DHCP service (**"$ systemctl restart dhcpd.service"**).

4.3.4 SNTP

For the SNTP service was necessary to install the **ntp** package. The SNTP service provides time information to devices connected to the network. This information can be used for time synchronization. This service is provided in the internal network and in the network between the IDS interface and the router (for time sync of the router). The configuration file is located in */etc/ntp.conf*. As is shown in Figure 4.12 it is necessary to modify the networks where the service will be provided and the time servers. This time servers are used to obtain information about the actual time.

```
...
restrict 192.168.123.0 mask 255.255.255.0 nomodify notrap
restrict 192.168.30.0 mask 255.255.255.0 nomodify notrap
...
server 0.cz.pool.ntp.org
server 1.cz.pool.ntp.org
server 2.cz.pool.ntp.org
server 3.cz.pool.ntp.org
...
```

Figure 4.12: Modifications in the SNTP configuration file

4.3.5 NetFlow

The NetFlow Collector is located in the internal network (IP address 192.168.123.1/24). The Collector is listening on port 9996. The software used to collect NetFlow records is the NetFlow Analyzer 11[2]. To install NetFlow Analyzer 11 it is required to install Java SDK. The software (NetFlow Analyzer) collects NetFlow records and also provides a web interface for visualization and data analyzing. Figure 4.13 shows several steps of the installation process. Represented by red color is the installation directory where the analysis software will be installed. In this case the default path will be */opt/ManageEngine/NetFlow*. In this directory are placed all binary files and scripts for execution of this software. These scripts are placed in the *./bin* directory. To run this software it is necessary to execute the script **run.sh** in this directory. The number 9996 in blue color represents the port, where the NetFlow Collector will be listening. The number 8080 in orange color is the port of the web server. To access the GUI when the NetFlow analyzer is running, it is necessary to type in the web browser the address *http://localhost:8080*. The default login and password required to access the application are „admin“, „admin“.

4.3.6 IDS and IPS

The IDS/IPS software used in this project was Snort. Regarding the installation and configuration, there are several documents[3] on the Snort project website.

The first step of the installation is to install the dependencies. The dependencies required for this software are **gcc, flex, bison, zlib, zlib-devel, libpcap, libpcap-devel, pcre, pcre-devel, libdnet, libdnet-devel** and **tcpdump**.

The next required step is to install the Snort packages. Several installation packages are provided depending on the type of the operating system. In this scenario it was used

[2]For more information: https://www.manageengine.com/products/netflow/.
[3]Useful documents for Snort installation/configuration: https://www.snort.org/documents.

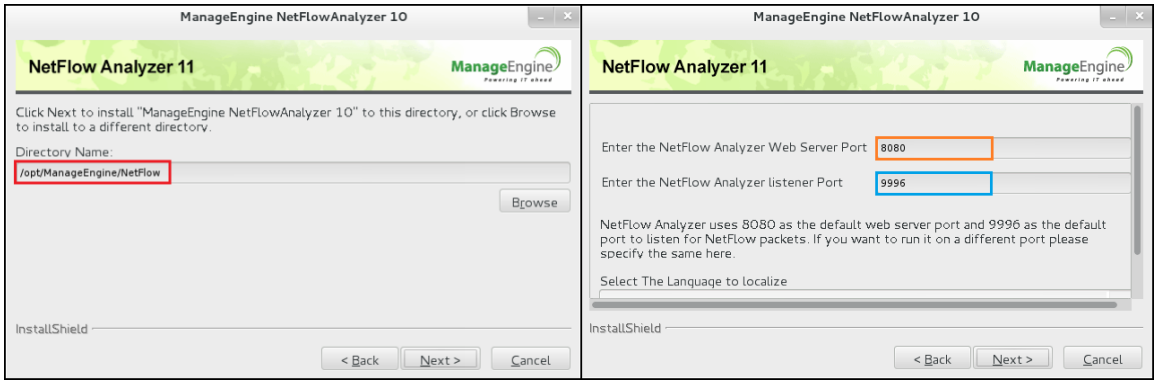


Figure 4.13: NetFlow Analyzer installation

packages for CentOS 7. The RPM packages necessary for the installation are **daq**[4] and **snort**[5]. It is possible to install them by the **yum** command. The configuration file of the Snort installation is placed in */etc/snort/snort.conf*. In this file is possible to specify the IP addresses of the devices in the network that provide services (DNS, FTP, SQL, etc.), home network, external network, monitoring rules, etc. An example of the Snort configurations is shown in Figure 4.14.

```
...
ipvar HOME_NET 192.168.123.0/24
ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET
...
```

Figure 4.14: Snort configurations in *snort.conf* file

In the same file it is also necessary to set rules. It is possible to obtain rules from several sources (paid rules, basic rules provided by Snort project or free to use rules). In this project were used rules[6] under GLPv2 license and subscribed rules provided by the Snort project. The rules need to be placed in the */etc/snort/rules* directory. It is possible to find in an additional file all additional options (*include* < *option* >)[7]. These *includes* need to be copied to the *snort.conf* file.

[4]Daq package: https://www.snort.org/downloads/snort/daq-2.0.5-1.centos7.x86_64.rpm.
[5]Snort package:https://www.snort.org/downloads/snort/snort-2.9.7.3-1.centos7.x86_64.rpm.
[6]Snort rules: https://rules.emergingthreats.net/open/snort-2.9.0/emerging.rules.tar.gz.
[7]Additional config. file: https://rules.emergingthreats.net/open/snort-2.9.0/emerging.conf.

Chapter 5

Security incidents

5.1 Reconnaissance attack

In this experiment will be shown an attack by **nmap** application for scanning open ports of the server placed in the internal network. The basic topology was slightly modified, as shown in Figure 5.1. The attacker is placed in the 192.168.30.0/24 network. This network is used for transferring traffic between the internal network and the router. The IDS is running on the interface with IP address 192.168.30.253.

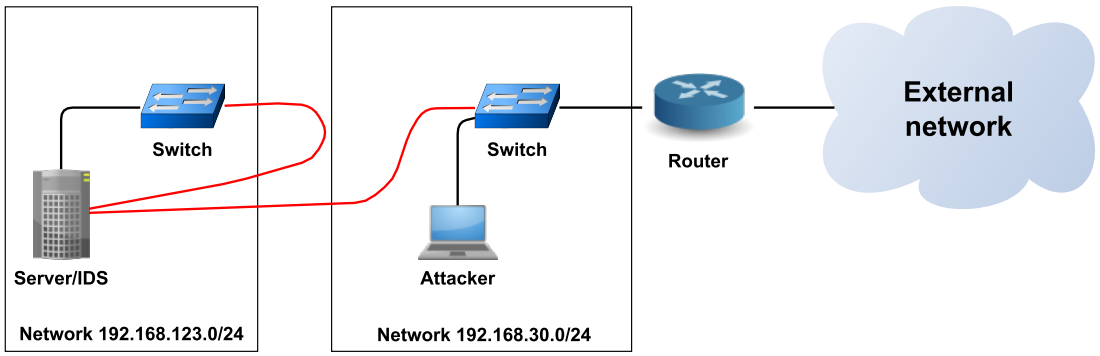


Figure 5.1: Topology of the experiment

This type of attack is very difficult to recognize. The maximum bandwidth captured in the internal network was not more than 80KB/s. Figure 5.2 shows the bandwidth of the whole experiment. Thus, the bandwidth cannot be used for attack recognition.

A better way to recognize the attack is to monitor the captured traffic in the network. The Nmap application has an unique sequence of messages that are sent to the target. It makes the recognition easier.

5.1.1 Traffic analysis

In this experiment the attacker used a TCP SYN packet for port recognition. The technique behind this attack is to not open a full TCP connection. The Nmap application will just send a SYN packet and wait for a response. Using this packet the attacker asks if the port (where the packet was sent) is open or not. The victim can answer with two types

of messages. The first type is a SIN/ACK message (it can also be SYN without ACK), representing an open port in the victim's device. The second one is a RST message, that indicates a non listening port. The port is also marked as filtrated if no answer is received, or if ICMP unreachable error is received. This description is valid only for TCP SYN scan. Attacks based on other techniques have also different packet types and sending orders. [13].

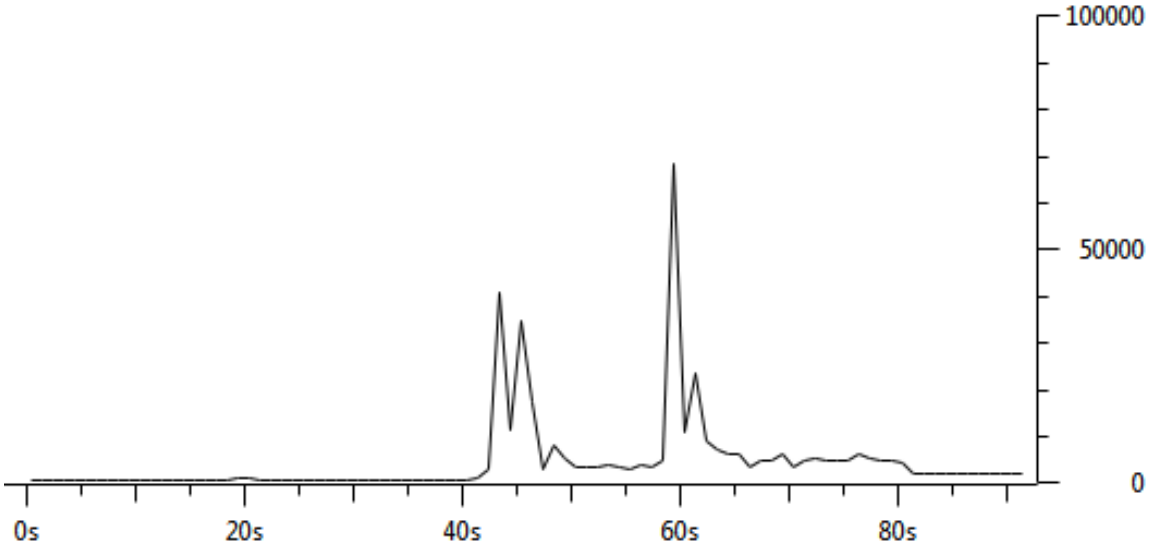


Figure 5.2: Bandwidth of the server during the attack

Figure 5.3 shows part of the captured traffic from the experiment. It is shown the 5900 port scanning, which is commonly used as a listening port for servers providing VNC service [23]. This communication consists of two ACK messages where the attacker scans the port and the victim responds (RST message), indicating a non listening port.

No.	Time	Source	Destination	Protocol	Length	Info
80	43.408485	192.168.30.60	192.168.123.1	TCP	60	36938→5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
87	43.408803	192.168.30.254	192.168.123.1	TCP	60	36938→5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
88	43.408824	192.168.123.1	192.168.30.254	TCP	54	5900→36938 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figure 5.3: Negative answer of nmap scanning

An example of the traffic is shown in Figure 5.4 where there is a communication (between the attacker and the victim), indicating that port 22 is listening (SSH service)[1]. The attacker sent a SIN message and received a SIN/ACK response from the victim. The attacker subsequently answered with a RST message. This message canceled the TCP negotiation.

No.	Time	Source	Destination	Protocol	Length	Info
79	43.408479	192.168.30.60	192.168.123.1	TCP	60	36938→22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
895	44.812152	192.168.123.1	192.168.30.60	TCP	60	22→36938 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
897	44.812189	192.168.30.60	192.168.123.1	TCP	60	36938→22 [RST] Seq=1 Win=0 Len=0

Figure 5.4: Positive answer of nmap scanning

[1]IANA: http://www.iana.org.

5.1.2 IDS analysis

IDS devices provide real time monitoring. This type of device monitors traffic, creates alerts and log files. Using the alerts is very easy to identify the type or the source of the attack. An example of a Snort alert is shown in Figure 5.5. This Figure shows an alert of a potential VNC attack. Alerts consist of the names of potential intrusions, priorities, IP addresses, port numbers, etc. All the traffic is analyzed using rules. In this experiment the rules did not cover all the scans. It means that malicious traffic was evaluated as normal traffic or rules did not recognize all the malicious activity. It is possible to change these rules or add custom rules, depending on the network topology or the administrator requirements. These modifications require detailed analysis of the attack.

```
...
[**] [1:2002910:5] ET SCAN Potential VNC Scan 5800-5820 [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/22-10:39:52.239341 192.168.30.254:36939 -> 192.168.123.1:5815
TCP TTL:58 TOS:0x0 ID:7937 IpLen:20 DgmLen:44
******S* Seq: 0x94AFDCC7 Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://doc.emergingthreats.net/2002910]

[**] [1:2002910:5] ET SCAN Potential VNC Scan 5800-5820 [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/22-10:39:52.341856 192.168.30.60:36940 -> 192.168.123.1:5815
TCP TTL:45 TOS:0x0 ID:38641 IpLen:20 DgmLen:44
******S* Seq: 0x94ACDCC4 Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://doc.emergingthreats.net/2002910]
...
```

Figure 5.5: Example of a snort alert of a potential VNC scan

Figure 5.6 shows all Snort log records captured during the experiment. Using these records is possible to realize the attack quicker and perform a more effective analysis of potential attacks. The logged records (unlike the captured traffic by tcpdump) are filtrated, with only necessary packets being stored in the file.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.30.60	192.168.123.1	TCP	60	36938→3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2	0.000176	192.168.30.254	192.168.123.1	TCP	60	36938→3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3	0.053442	192.168.30.60	192.168.123.1	TCP	60	36938→1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.063930	192.168.30.60	192.168.123.1	TCP	60	36938→5432 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	0.070986	192.168.30.60	192.168.123.1	TCP	60	36938→5907 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.073024	192.168.30.254	192.168.123.1	TCP	60	36938→5904 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	1.791897	192.168.30.254	192.168.123.1	TCP	60	36939→5815 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	1.894412	192.168.30.60	192.168.123.1	TCP	60	36940→5815 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	2.191419	192.168.30.60	192.168.123.1	TCP	60	36939→5432 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10	2.191887	192.168.30.254	192.168.123.1	TCP	60	36939→5432 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	2.294636	192.168.30.60	192.168.123.1	TCP	60	36940→5432 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	2.296641	192.168.30.254	192.168.123.1	TCP	60	36940→5432 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	14.224152	192.168.30.60	192.168.123.1	TCP	60	36939→1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	14.224373	192.168.30.254	192.168.123.1	TCP	60	36939→1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
15	14.386759	192.168.30.60	192.168.123.1	TCP	60	36940→1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	14.386862	192.168.30.254	192.168.123.1	TCP	60	36940→1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	33.057375	192.168.30.60	192.168.123.1	TCP	60	36939→1521 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
18	33.057703	192.168.30.254	192.168.123.1	TCP	60	36939→1521 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Figure 5.6: Captured packets by Snort

The result of the attack (what is visible to the attacker) is the list of ports used by the server. Using this result the attacker can create a map of the network topology and

prepare an intrusion to this topology. In the Figure 5.7 are shown statistics from the nmap application achieved with the experiment.

```
hugo@hugo-virtual-machine:~$ sudo nmap  192.168.123.1

Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-21 17:37 CEST
Nmap scan report for 192.168.123.1
Host is up (0.0012s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
3306/tcp open  mysql
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 38.69 seconds
```

Figure 5.7: Data acquired by the attacker

5.2 Malware

The second experiment consists of monitoring malware activity in the network topology. The infected victim machine has IP address 192.168.123.52/24. The malware[2] used in this project at first sight looks like a porn movie with the name **ZeroAccess_xxx_porn_--movie.avi.exe**. After the execution it tries to cover its identity behaving like Adobe Flash Player. The execution starts the download process of additional malware sources for later malicious activity. Figure 5.8 shows a pop-up window that displays the immediate execution.

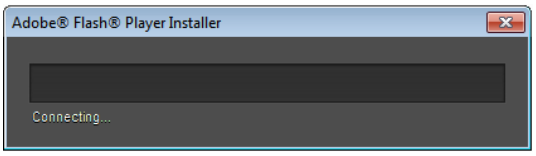


Figure 5.8: Malware pop-up window

When the download activity is finished, the malicious software informs a connection problem and ask for a new execution. After receiving a confirmation by the user, the download process message is shown again. An example of the message is shown in Figure 5.9.

5.2.1 Analysis using Netflow Analyzer

During the experiment were used several monitoring systems. The Netflow Analyzer provides several information, useful for faster analysis (bandwidth protocol usage).

The bandwidth captured by the router during the experiment is shown in Figure 5.10. It is possible to notice in the figure that the maximum bandwidth of the application was

[2]Source of the malware software: https://github.com/ytisf/theZoo.

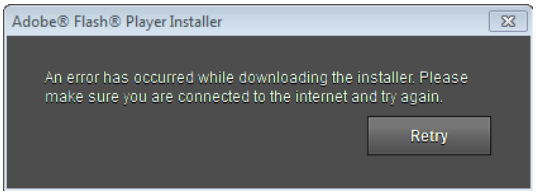


Figure 5.9: Malware warning message

around 712,5 KB/s. The maximum was reached during the download phase. From this amount of data transferred it is possible to realize the presence of some malicious software.

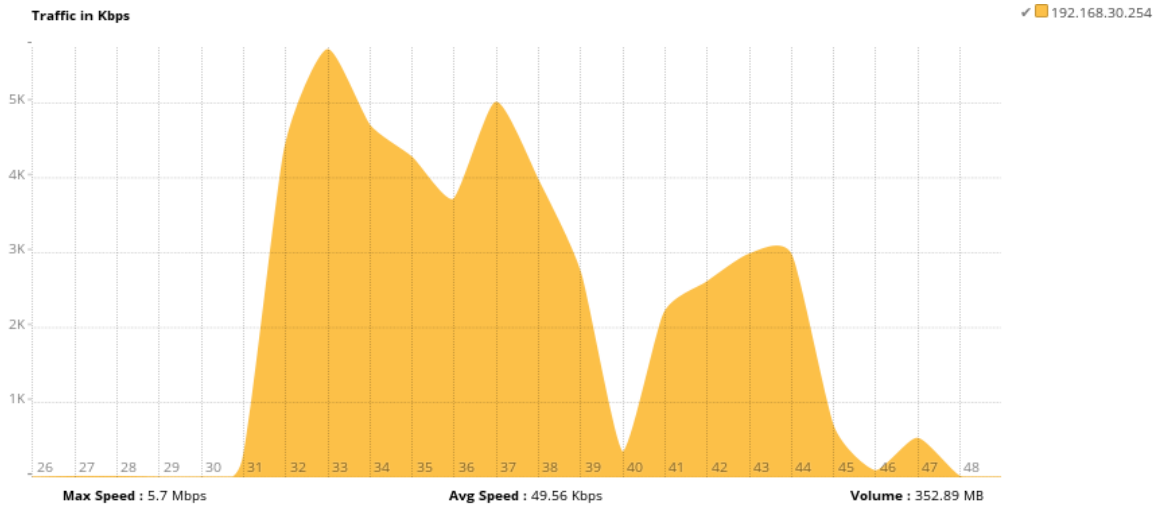


Figure 5.10: Malware bandwidth

Statistics regarding application usage are shown in Figure 5.11. It is possible to notice that the biggest amount of communication was made by HTTP protocol. It indicates that the malicious software was using some HTTP mirror server to download resources.

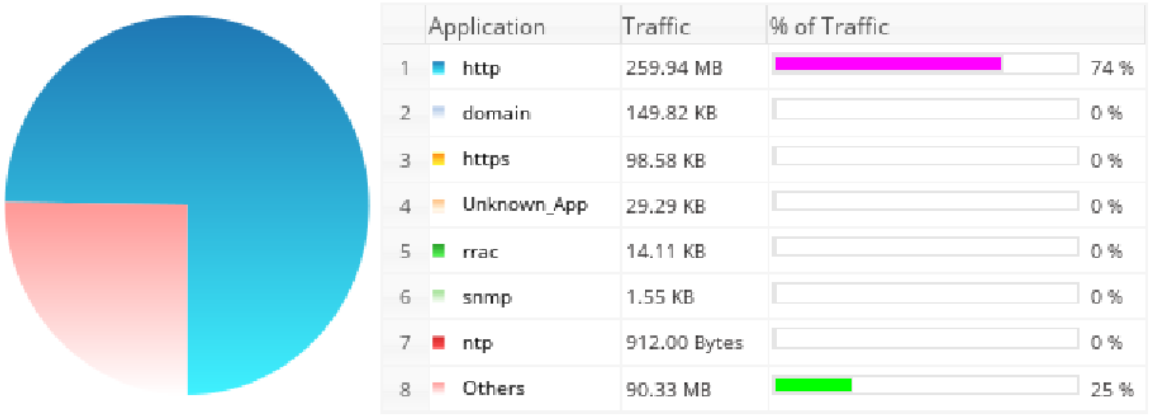
During the experiment the most used destination address was 192.168.123.52. It means that the malicious software was attacking only the victim's machine and it did not try to distribute itself to other machines. Thus, the malicious software was not a worm. Figure 5.12 shows the statistics of the top destination addresses captured by the NetFlow Collector.

The biggest amount of data was sent from the IP address 2.16.190.8. It is possible to consider that this IP address belongs to the main communication channel of the malicious software. Figure 5.13 shows the statistics of the top source IP addresses analyzed by the NetFlow Collector.

It is possible to use the data from the NetFlow Analyzer for offline traffic analysis. The obtained information (source addresses, destination addresses) can be useful to find the source of the attack faster.

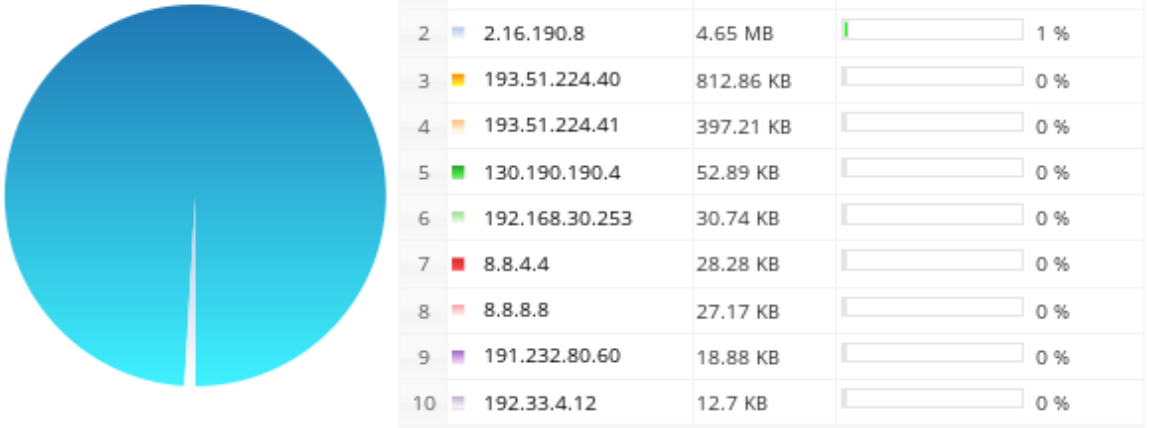
5.2.2 Traffic analysis

The offline traffic analysis can help deepen the understanding about the malware software activity. The basic information used for analysis was the source addresses captured by



	Application	Traffic	% of Traffic
1	http	259.94 MB	74 %
2	domain	149.82 KB	0 %
3	https	98.58 KB	0 %
4	Unknown_App	29.29 KB	0 %
5	rrac	14.11 KB	0 %
6	snmp	1.55 KB	0 %
7	ntp	912.00 Bytes	0 %
8	Others	90.33 MB	25 %

Figure 5.11: Top application usage report



	Destination	Traffic	% of Traffic
1	192.168.123.52	383.55 MB	>100 %
2	2.16.190.8	4.65 MB	1 %
3	193.51.224.40	812.86 KB	0 %
4	193.51.224.41	397.21 KB	0 %
5	130.190.190.4	52.89 KB	0 %
6	192.168.30.253	30.74 KB	0 %
7	8.8.4.4	28.28 KB	0 %
8	8.8.8.8	27.17 KB	0 %
9	191.232.80.60	18.88 KB	0 %
10	192.33.4.12	12.7 KB	0 %

Figure 5.12: Top destination addresses report

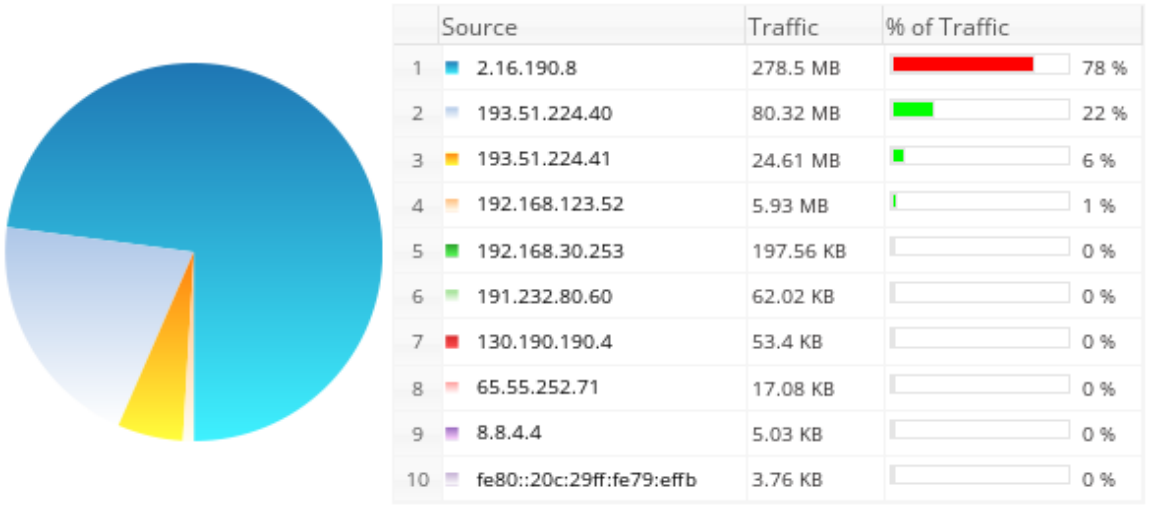
the NetFlow Collector. The first analyzed traffic was the HTTP packets, because previous analysis shows that the malware was probably communicating with a HTTP mirror server.

After analysis of the HTTP headers, it was possible to notice that the traffic belonging to the IP addresses 193.51.224.40 and 193.51.224.41 was related to Windows updates. During the experiment it had a traffic of more than 100 MB.

Figure 5.14 shows the initialization of a TCP connection and a HTTP GET request to download additional malicious software. The communication took place between the victim device (192.168.123.52) and the server where the malicious software is stored (2.16.190.8).

A detailed view of the HTTP header of the packet is shown in Figure 5.15. The header consists of a Request URI and a full request message. These fields and the destination IP address are marked by red color.

After the installation phase of the malware activity it was detected a weird traffic consisting of UDP packets. The destination addresses, according to the tool **whois**, were several organizations around the world (especially countries where tracking or persecuting



	Source	Traffic	% of Traffic
1	2.16.190.8	278.5 MB	78 %
2	193.51.224.40	80.32 MB	22 %
3	193.51.224.41	24.61 MB	6 %
4	192.168.123.52	5.93 MB	1 %
5	192.168.30.253	197.56 KB	0 %
6	191.232.80.60	62.02 KB	0 %
7	130.190.190.4	53.4 KB	0 %
8	65.55.252.71	17.08 KB	0 %
9	8.8.4.4	5.03 KB	0 %
10	fe80::20c:29ff:fe79:effb	3.76 KB	0 %

Figure 5.13: Top source addresses report

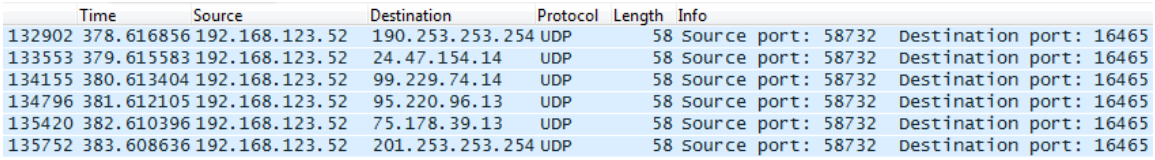
No.	Time	Source	Destination	Protocol	Length	Info
466	143.970022	192.168.123.52	2.16.190.8	TCP	66	49208→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
467	144.011089	2.16.190.8	192.168.123.52	TCP	60	80→49208 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
468	144.011198	192.168.123.52	2.16.190.8	TCP	54	49208→80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
469	144.011422	192.168.123.52	2.16.190.8	HTTP	235	GET /get/flashplayer/update/current/install/install_all_win_cab
470	144.012363	2.16.190.8	192.168.123.52	TCP	60	80→49208 [ACK] Seq=1 Ack=182 Win=64240 Len=0
471	144.063866	2.16.190.8	192.168.123.52	TCP	303	[TCP segment of a reassembled PDU]

Figure 5.14: Initialization of the malware activity

```
⊞ Frame 469: 235 bytes on wire (1880 bits), 235 bytes captured (1880 bits)
⊞ Ethernet II, Src: Vmware_99:7f:ab (00:0c:29:99:7f:ab), Dst: Vmware_fa:5f:52 (00:0c:29:fa:5f:52)
⊞ Internet Protocol Version 4, Src: 192.168.123.52 (192.168.123.52), Dst: 2.16.190.8 (2.16.190.8)
⊞ Transmission Control Protocol, Src Port: 49208 (49208), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 181
⊟ Hypertext Transfer Protocol
  ⊟ GET /get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z HTTP/1.1\r\n
    ⊟ [Expert Info (Chat/Sequence): GET /get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z HTTP/1.1\r\n]
        [GET /get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z
      Request Version: HTTP/1.1
    User-Agent: Flash Player Seed/3.0\r\n
    Host: fpdownload.macromedia.com\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [Full request URI: http://fpdownload.macromedia.com/get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z]
    [HTTP request 1/7]
    [Response in frame: 32580]
    [Next request in frame: 32855]
```

Figure 5.15: URI request for downloading additional malicious software

the attacker is impossible). For example, the address 190.253.253.254 belongs to COLOMBIA TELECOMUNICACIONES S.A. ESP and the address 95.220.96.13 belongs to Net By Net Holding LLC (IP address registered in Russia). Almost all UDP traffic communicates with the same destination port, which is 16465. Figure 5.16 shows an example of the traffic.



	Time	Source	Destination	Protocol	Length	Info
132902	378.616856	192.168.123.52	190.253.253.254	UDP	58	Source port: 58732 Destination port: 16465
133553	379.615583	192.168.123.52	24.47.154.14	UDP	58	Source port: 58732 Destination port: 16465
134155	380.613404	192.168.123.52	99.229.74.14	UDP	58	Source port: 58732 Destination port: 16465
134796	381.612105	192.168.123.52	95.220.96.13	UDP	58	Source port: 58732 Destination port: 16465
135420	382.610396	192.168.123.52	75.178.39.13	UDP	58	Source port: 58732 Destination port: 16465
135752	383.608636	192.168.123.52	201.253.253.254	UDP	58	Source port: 58732 Destination port: 16465

Figure 5.16: Malware network activity

5.2.3 IDS analysis

In this experiment were used rules provided by the Snort project under subsciption. An IDS station was running on the interface with IP address 192.168.30.253. During the experiment were captured several alerts and packets for analysis.

As shown in Figure 5.17 the IDS monitoring detects several incidents. The incidents were classified as Trojan Horse (type ZeroAccess) outbound communication. During the experiment were also captured around 600 packets for detection by the Snort IDS. Thanks to this monitoring system it is possible to recognize not only incoming attacks but also infected machines inside the monitoring network.

```
...
[**] [1:23493:5] MALWARE-CNC Win.Trojan.ZeroAccess outbound communication [**]
[Classification: A Network Trojan was Detected] [Priority: 1]
07/26-15:39:50.658445 192.168.123.52:58732 -> 184.253.253.254:16465
UDP TTL:128 TOS:0x0 ID:23612 IpLen:20 DgmLen:44 DF
Len: 16
[Xref => http://www.virustotal.com/file/50cdd9f6c5629630c8d8a3a4fe7
d929d3c6463b2f9407d9a90703047e7db7ff9/analysis/]

[**] [1:23493:5] MALWARE-CNC Win.Trojan.ZeroAccess outbound communication [**]
[Classification: A Network Trojan was Detected] [Priority: 1]
07/26-15:39:51.656702 192.168.123.52:58732 -> 190.253.253.254:16465
UDP TTL:128 TOS:0x0 ID:23613 IpLen:20 DgmLen:44 DF
Len: 16
[Xref => http://www.virustotal.com/file/50cdd9f6c5629630c8d8a3a4fe7
d929d3c6463b2f9407d9a90703047e7db7ff9/analysis/]
...
```

Figure 5.17: Snort malware detection alerts

Chapter 6

Conclusion

6.1 Summary

In this paper was shown the configuration of a network topology and the analysis of network incidents from network traffic. The network topology presents several monitoring tools. These tools can be used to exposure computer attack/intrusions or malware activities (worms, Trojan horses, etc.). Several experiments show how to use these tools to uncover these malicious activities in the network. It was also developed several configuration files to make it easier to configure the monitoring station.

6.2 Improvements

There are several possible improvements related to this work. Probably one of the most interesting ones would be a detailed software analysis of the malware software (binary data analysis). This improvement would help find weak points of the network topology and also design mechanisms to avoid an intrusion like that. There is also other alternative IDSs. Another improvement could be the comparison of results and rules from Snort and Suricata (another IDS). This comparison can also include practical and economical analysis of these IDSs.

Bibliography

[1] Admin of ComputerNetworkingNotes.com. Network security types of attacks. *ComputerNetworkingNotes.com*, 2014. [online]. [cit. 2015-01-31].

[2] Usha Banerjee, Ashutosh Vashishtha, and Mukul Saxena. Evaluation of the capabilities of wireshark as a tool for intrusion detection. *International Journal of Computer Applications*, 6(7), 2010.

[3] David Barroso Berrueta. A practical approach for defeating nmap os- fingerprinting. *Retrieved March*, 12:2009, 2003.

[4] Ida Mae Boyd. The fundamentals of computer hacking. *SANS Institute*, 2000.

[5] Benoit Claise. Cisco systems netflow services export version 9. 2004.

[6] Leslie Daigle. Whois protocol specification. 2004.

[7] P.ARUNA DEVI, S.RANI LASKHMI, and K.SATHIYAVAISHNAVI. A study on network security aspects and attacking methods. *International Journal of P2P Network Trends and Technology-Volume 3 Issue 2-2013*, 2013. [online]. [cit. 2015-01-28].

[8] Paul Ferguson. Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing. 2000.

[9] Nick Ierace, Cesar Urrutia, and Richard Bassett. Intrusion prevention systems. *Ubiquity*, 2005(June):2–2, 2005.

[10] SANS Institute. Understanding intrusion detection systems. *SANS Institute Reading Room site.*, 2001.

[11] Atul Kahate. *Cryptography and Network Security, Third edition.* McGraw Hill Education, 2013.

[12] Jonathan Lemon et al. Resisting syn flood dos attacks with a syn cache. In *BSDCon*, volume 2002, pages 89–97, 2002.

[13] Gordon Fyodor Lyon. *nmap(1) - Linux man page.*

[14] ManageEngine. Netflow analyzer. https://www.manageengine.com/products/netflow/. Accessed: 2010-07-20.

[15] Robert Moskovitch, Yuval Elovici, and Lior Rokach. Detection of unknown computer worms based on behavioral classification of the host. *Computational Statistics & Data Analysis*, 52(9):4544–4566, 2008.

[16] Nicholas Pappas. Network ids & ips deployment strategies. *SANS Institute*, 2008. [online]. [cit. 2015-07-20].

[17] Orbit-Computer-Solutions. Types of network attacks. *Orbit-Computer-Solutions.Com*, 2013. [online]. [cit. 2015-01-28].

[18] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki. The internet protocol journal-volume 7, number 4 distributed denial of service attacks. *ISSUES*, 7(4), 2004.

[19] Malek Ben Salem and Salvatore J Stolfo. Decoy document deployment for effective masquerade attack detection. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 35–54. Springer, 2011.

[20] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps). *NIST special publication*, 800(2007):94, 2007.

[21] Ryan Spangler. Packet sniffer detection with antisniff. *University of Wisconsin, Whitewater. Department of Computer and Network Administration*, 2003.

[22] William Stallings. *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE, FIFTH EDITION*. Prentice Hall, 2011.

[23] Ariel Stolerman. Rfc 6143: The remote framebuffer (rfb) protocol analysis. 2013.

[24] Peter Szor. *The art of computer virus research and defense*. Pearson Education, 2005.

[25] Trustwave corporate. 2014 business password analysis. *Trustwave.com*, 2014. [online]. [cit. 2015-01-28].

List of Figures

Appendix A

CD Content

- scripts: configuration scripts of the CentOS server and the Mikrotik router
- rules: Snort rules
- pcap: captured traffic from experiments
- log: log files from IDS
- malware: malware software used in the experiments
- doc: latex source files

Appendix B

DNS zone files

Zone file leopard.org (*/var/named/leopard.org*):

```
$TTL 1D
@ IN SOA zeus.leopard.org. root.leopard.org. (
       2014071001 ;Serial
       3600       ;Refresh
       1800       ;Retry
       604800     ;Expire
       86400      ;Minimum TTL
)

       IN    NS   zeus.leopard.org.
       IN    A    192.168.123.1

;srvce.port.owner-name ttl class rr pri weight port target
_http._tcp.leopard.org.   IN   SRV 0   5   80   www.leopard.org.
_http._udp.leopard.org.   IN   SRV 0   5   80   www.leopard.org.

zeus   IN   A   192.168.123.1
www    IN   A   192.168.123.1
dns    IN   A   192.168.123.1
ns     IN   A   192.168.123.1
```

Reverse zone file 123.168.192.db (*/var/named/123.168.192.db*):

```
$TTL 86400
@ IN SOA zeus.leopard.org. root.leoaprd.org. (
       2014071001 ;Serial
       3600       ;Refresh
       1800       ;Retry
       604800     ;Expire
       86400      ;Minimum TTL
)

       IN   NS   zeus.leoaprd.org.

; define IP address and hostname
1   IN   PTR   zeus.leopard.org.
1   IN   PTR   www.leopard.org.
1   IN   PTR   ns.leoaprd.org.
1   IN   PTR   dns.leopard.org.
```