
**ANALYSIS AND EVALUATION OF NETWORK INTRUSION
DETECTION METHODS; A CASE OF ANOMALY
DETECTION AND SIGNATURE DETECTION
APPROACHES**

**BY
KAWEESA JAMES
BCS/2389/71/DU**

**PROJECT REPORT SUBMITTED TO THE SCHOOL OF
COMPUTER STUDIES IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE AWARD OF
BACHELOR OF COMPUTER SCIENCE
OF KAMPALA INTERNATIONAL
UNIVERSITY**

JUNE, 2010

DECLARATION

I, **Kaweesa James**, hereby declare to the best of my knowledge that the work embodied in this research paper is my original work arrived at through reading and research and has not been published or submitted to any University or any other Institution of Higher Learning for any academic award." The literature and citation from other scholars' work have been fully referenced and acknowledged in the text and bibliography.

Signature.....

Date 02-July-2010

Kaweesa James

(Student)

APPROVAL

This research dissertation has been submitted for examination with my approval as the
University Supervisor:

Signature.....

Date

Ssegawa E. James Kiggundu

DEDICATION

I dedicate this work to my mother M/s Namukwaya Alice who laid my education foundation, Mr Luigi Dante, my brothers and sisters. Your love, care and moral support made it possible for me. God be with you.

ACKNOWLEDGEMENT

First I would like to thank Jehovah God for his favor, divine protection, mercy, love, provision, grace and his anointing upon my life. Am very grateful to my lovely, caring, sweet and dearest mother M/s. Alice Namukwaya, Mr.Luigi Dante for their patience and support throughout my education.

I wish to thank my supervisor Mr. Segawa James for his ideas and professional advice encoded to me through out the research period much of whatever integrity and quality of this report has is a direct result of his proper guidance and support.

Am equally obliged to thank my sisters, brothers and relatives who never abonded me during the difficult times they have been always on my side to see me succeeding in my education. I want to acknowledge the support of all my friends for the support rendered to me in one way or the other my the almighty reward exceedingly and abundantly

LIST OF TABLES

Table 1: 2.1-below provides definitions for these Alarm types.	15
Table 2: 4.1 –Experimental tools summary.....	37
Table 3: 5.1- training window experimental results.....	40

LIST OF FIGURES

Figure 1: 2.1- IDS architecture	12
Figure 2: 2.2- The main differences between HIDSs and NIDSs:	14
Figure 3: 2.3- Darpa experiment setup	20
Figure 4: 2.4-Trident framework.....	21
Figure 5: 3.1 -Training window experiment design	26
Figure 6: 3.2- Scenario experiment design.....	27
Figure 7: 4.1 WinPcap installation.	30
Figure 8: 4.2-creating snort database.....	31
Figure 9: 4.3-creating snort user account	31
Figure 10: 4.4 snort configuration file.....	32
Figure 11: 4.5 –Reconnaissance with Nmap step 1	34
Figure 12: 4.4 –Reconnaissance with Nmap step 2.....	35
Figure 13: 4.7 ftp server	36
Figure 14: 5.1–1 st attack SPADE detection Rate	39
Figure 15: 5.2– 2 nd attack SPADEdetection rate	39

LIST OF ACRONYMS

ISO	International Organization for Standardization
CCITT	International Telegraph and Telephone Consultative Committee
ITU-T	International Telecommunication Union –Telecommunication
IP	Internet Protocol
TCP	Transmission Control Protocol
IT	Information Technology
IDS	Intrusion Detection System
NIST	National Institute of standard and technology
Dos	Denial-of-service
DDoS	Distributed Denial-of-Service
SPADE	Statistical Packet Anomaly Detection Engine
NIDS	Network-based Intrusion Detection System
HIDS	Host-based Intrusion Detection System
DMZ	DeMilitarized Zone
DUT	Device Under Test
MIT	Massachusetts Institute of Technology
MAC	Media Access Control
IETF	Internet Engineering Task Force
MACE	malicious traffic composition environment
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol

ABSTRACT

Many Network administrators and network analysts in organizations do not know which intrusion detection system to use. This is partly due to the fact that there is no clear comparison between the different intrusion detection systems. Therefore, organizations need concrete comparisons between different tools in order to choose which best suite for their needs is. This research aims at comparing anomaly with signature detection methods in order to establish which is best suited to guard organization, such as data theft. The difference between anomaly and signature-based detection is that an anomaly Intrusion Detection System needs to be trained and generate many alerts, the majority of which being false alarms; hence another aim is to establish the influence of the training period length of an anomaly Intrusion Detection system on its detection rate. Hence, this research presents a Network-based Intrusion Detection System evaluation testbed setup, and it shows the setup for two of these using the signature detector (Snort) and the anomaly detector Statistical Packet Anomaly Detection Engine (SPADE). The evaluation testbed is then used to create a data theft scenario that includes the following stages: reconnaissance; gaining unauthorized access; and finally data theft. Therefore, it offers the opportunity to compare both detection methods with regards to that threat. This research acts as documentation for setting up a network Intrusion Detection System evaluation testbed. SPADE, lack a centralized documentation and no research paper could be identified that clearly documents the configuration of an evaluation testbed for Intrusion Detection System. Standards for evaluating Intrusion Detection System could not identified, and thus this required the creation of a bespoke evaluation testbed which, in turn, limited the time dedicated to evaluating the threat scenario itself. Along with this, results show that configuration, testing and verification of the anomaly detection system is highly error-prone.

TABLE OF CONTENT

DECLARATION.....	i
APPROVAL.....	ii
DEDICATION	iii
ACKNOWLEDGEMENT.....	iv
LIST OF TABLES	v
LIST OF FIGURES.....	vi
LIST OF ACRONYMS.....	vii
ABSTRACT	viii
TABLE OF CONTENT	ix
CHAPTER ONE.....	1
INTRODUCTION.....	1
1.0 Introduction	1
1.1 Background	1
1.2 Statement of the problem	2
1.3 Main objective.....	2
1.3.1 Specific objectives.....	2
1.4 Research questions	2
1.5 Scope	3
1.5.1 Geographical scope	3
1.5.2 Research Scope.....	3
1.6 Significance	3
1.7 Conceptual framework	4
CHAPTER TWO.....	5
LITERATURE REVIEW	5
2.1 Introduction	5
2.2 Security management	5
2.2.1 Definition.....	5
2.3 Computer systems threats.....	6

2.3.1 Information system threats	6
2.3.2 Threat applications	7
2.3.3 Threat consequences.....	8
2.4 Firewalls	9
2.4.1 Packet filters	10
2.4.2 Application layer gateway	11
2.4.3 Stateful packet filters.....	11
2.5 Intrusion detection system.....	12
2.5.1 Definition.....	12
2.5.2 Host-based IDS and Network-based IDS	13
2.5.3 Anomaly detection	15
2.5.4 Signature detection	16
2.5.5 Hybrid systems	17
2.6 Other security Tools	17
2.6.1 IDS related tools.....	17
2.6.2 Antivirus.....	18
2.7 Testing of Intrusion detection systems	18
2.7.1 Evaluation metrics for IDSs	18
2.7.2 Offline Evaluation	19
2.7.3 Online Evaluation.....	20
 CHAPTER THREE.....	 23
METHODOLOGY	23
3.0 Introduction	23
3.1 Research design.....	23
3.2 Target Population	23
3.3 Research instruments.....	24
3.4 Data presentation/analysis.....	24
3.5 Logical network architecture.....	25
3.6 Training window experiment	25
3.7 Scenario	26

3.8 Hypothetical results	27
CHAPTER FOUR	29
PHYSICAL NETWORK ARCTECTURE	29
4.1 Introduction	29
4.2 Background traffic generation	29
4.3 Intrusion detection system	29
4.3.1 Signature detection	29
4.3.2 Anomaly detection	32
4.4 Training window experiment	33
4.4.1 Experimental parameters	33
4.5 Scenario	34
4.5.1 Exploit generation	34
4.5.2 FTP server	36
4.5.3 Experimental parameters	37
CHAPTER FIVE	38
FINDINGS AND RECOMMENDATION	38
5.1 Introduction	38
5.2.2 Findings	40
5.3 Scenario	40
5.3.1 Results	40
5.3.2 Findings	41
5.4 Recommendation	42
5.4.1 Experiment	42
5.5 Area of further research	42
5.6 Conclusions	43

CHAPTER ONE

INTRODUCTION

1.0 Introduction

This chapter gives a background of the research that was carried out. The purpose of this background is to provide a description of what this research is all about, by showing the aim and objectives of the research, the scope of the research, and finally the significance.

1.1 Background

The world of Internet is ever expanding as more and more companies realize the financial and organizational benefits that come from having networks and Internet. Its use has spread to the core of most business. The progression of technology has facilitated this expansion in a number of ways which include; the growth of broadband Internet has meant that companies can extend their activity and increase productivity. Since the Internet developed recently at such a fast rate, its availability increased greatly. Connections to the Internet are now available anywhere and at relatively low prices. This means that virtually anybody can access any information. According to NIST, (2006), this ease of accessibility to resources introduced a new kind of criminality: cyber-crime, this type of crime developed exponentially during the past decade, mainly due to the democratization of the Internet. Data is the most important asset in an organization. This highlights the crucial need for network security in order to keep the data and system secure. Ingham and Forrest, (2002), states that computer network security is often deployed in two ways. The first security application tries to establish a strong outside barrier in order to prevent unauthorized users gaining access to the system, since internal users still need to access resources outside the local network, this barrier has let some communications go through. Intruders usually take advantage of these characteristics to carry out exploits.

According R. Bace and P. Mell, (2001), the second type of security applied is monitoring the network for traces of exploits. Tools to achieve computer network security are numerous and often organizations do not know what to invest in. This thesis aims to

address this issue by providing a direct comparison between a signature intrusion detection system (IDS) and anomaly IDS in order for organization to chose the proper technology to mitigate for identity theft.

1.2 Statement of the problem

Most organizations have failed to protect their systems and applications that support the business processes from known and unknown attacks before they disrupt any business. Most often companies go online when they have not enforced security on computer system or network, yet there is a numerous growth of sophisticated hacker tools at a faster rate. In addition to that workers of companies still share resources using well known exploits, trust relationships and default settings, thus they remain prone to these threats, yet most companies do not give priority to network security in their budget. These networks left unplanned lead to heavy financial losses resulting from intrusion and inefficient resource utilization such as bandwidth waste through unwarranted server request from network nodes. Deploying intrusion prevention techniques such as a firewall and anti virus may not always be effective.

1.3 Main objective

The overall aim of this project is to provide performance analysis and evaluation of Network-based Intrusion Detection System (NIDS).

1.3.1 Specific objectives

In order to achieve the main objective, there are intermediary objective to achieve.

- i. Investigate current IDS evaluation methodologies
- ii. Establish the availability of tools required to create an evaluation test bed.
- iii. Carry out a critical appraisal of network security configuration
- iv. Setup a testbed for analysis and evaluation of IDS

1.4 Research questions

- i. What kind of methods used in evaluating IDS?
- ii. What kind of tools used in analysis and evaluation of IDS?

- iii. What kind of network security configuration used by organizations?
- iv. What kind of testbed needed for analysis and evaluation of IDS?

1.5 Scope

1.5.1 Geographical scope

During this research two organizations were visited, Canadian physical aid and relief (CPAR) on plot 3302 Diplomat Zone, Kasanga Gaba Rd and national agricultural advisory services (NAADS) on plot 346 Nakasero road, Mukwasi house . System administrators from these organizations were interviewed using the interview guide questions as indicated in appendix A.

1.5.2 Research Scope

Analyzing and evaluating Network intrusion detection systems involves several methods and different types of Network intrusion detection systems; but this research is going to be restricted to analyzing and evaluating anomaly based detection system and signature based detection system in order to expose their weakness and strength in handling identity theft.

1.6 Significance

The significance of this project is reflected in the area where it will be able to solve a series of challenges that companies or IT professional face in choosing the best Intrusion detection system. Thus when it is completed, it will be capable of accomplishing the following tasks: -

Network administrator or system administrator will be able to choose the best IDS that will provide a working environment free from threats and without disruptive messages and this will help reduce on bandwidth utilization. Well planned intrusion detection will also simplify network management. As the network expands, combining different techniques gives a better coverage and more effective intrusion detection and hence

prevention. Identity and data theft will be minimized, due to the fact that NIDS will be able to decide which information should enter or leave the system.

In the long run, network managers or system administrators will engage in other productive endeavors.

1.7 Conceptual framework

This report discusses a study that deals with Intrusion Detection Systems (IDSs) in computer networks. IDSs are systems that automate the process of monitoring and analysis the events occurring in a computer system or network, analyzing them for signs of security breaches. As network attacks have increased in number and severity over the past few years, IDSs have become a necessary addition to the security infrastructure of most organizations. As the technology move into the new frontier of Internet, IDSs have become a vital need to secure organization network which is unsecured in nature. This research will study on analysis and evaluation of network intrusion methods. Most of network administrators and users have problem with current intrusion detection mechanisms which are not flexible enough to provide early detection of intruders in networks. Intrusion detection is stated as critical, but reactive function. An improvement beyond intrusion detection to intrusion protection that adds the proactive pieces around the core function of intrusion detection is a must. Therefore, the objectives of this study are to analyze the current framework of IDS, to propose a conceptual framework of proactive IDS and to validate the framework by using the prototype which focuses on setting a testbed for signature and anomaly detection methods with background traffic and exploit generations as attack figure . This proposed framework also introduces a new mechanism that is proactive function, which is very useful in preventing the network threats. This framework can be a good practice to future researchers in providing proper proactive IDS. At the mean time, this framework will assist administrator in gaining a clear understanding on how to choose the best IDS that can be implemented in network environment.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter is aimed at reviewing related literature in order to give an insight into the research as well as showing that there is a need for this research. Thus literature reviewed from text books and papers that deals with computer network security

2.2 Security management

Before discussing security devices and computer systems threats we need to have an overview of what information system security involves. This part defines security management and outlines the computer network security threats to organizations.

2.2.1 Definition

According to Siponen and Oinas Kukkonen, (2007), security management is a mean of maintaining a secure information system in companies, including information system planning and evaluation. Security management should cover design, implementation and testing of processes and devices aiming at keeping secure company's asset and keeping at a minimum security risk performance analysis of network based forensic system for in-line and out-of-line detection and log. In the past, information security involved the security department and network management side of an organization, and the main problem was that both sides were not always clearly defined or present in an organization and did not interact with each other and unaware of each other. The security department was producing security technologies such as software or hardware dedicated to one task in order to solve specific challenges, while the other department was trying to standardize management solutions.

Bake and Wallace, (2007), shows that such an approach cannot work since information security is not limited to technical problems. Indeed, they show that the close relationship

between technology and business functionality has always been the source of costly information security incidents.

According to Hale and Brusil (2007), security was considered as a risk management however now day most companies recognize it as a competitive and economic advantage

2.3 Computer systems threats

This part discusses computer system threats and vulnerabilities and presents a classification of threats, followed by different example of possible application of threats. It also highlights the consequences of successful computer system intrusions.

2.3.1 Information system threats

Newman, (2006), defines a threat as any potential occurrence, either accidental or malicious, that can have an undesirable effect on assets and resources of the organization. Threats can have a harmful effect by trying to break the three main goals of computer security include, breach of confidentiality gaining access to private information, breach of integrity tampering with or accessing private information, breach of availability disrupting access to information or service. Intrusions can also be sub-classified into two categories, active and passive intrusions. Active intrusions involve direct action on data, resource or hardware whereas passive intrusions do not interfere directly in computer system. As well as passive attacks can be classified as insider attack or outsider attack, provided the source of attack being from inside the targeted network or out side from it. Computer intrusions generally follow the following stages, covering tracks: hide traces of intrusion, maintaining access: escalate privileges, gaining access: take control of a user account, reconnaissance: collection of information (structure of network, equipment, etc) about the target system, access to equipment, scanning: scan for vulnerabilities to exploit

NIST, (2006), provide a list of threats categories which sums up the different types of existing threats they include, Software flaws and configuration errors-the main source of vulnerabilities within computer systems. The fact that software programmers give priority

to functionality rather than security, Leads to common program flaws, such as buffer overflows, which are the most common source of exploit, the second threat is brute force attack this aims at gaining unauthorized access this type of attack tries all the possible combinations of a password for a given username in order to gain access to the corresponding account, the third is file alteration this breaches the integrity characteristic of computer security. It involves changing data in a data collection or changing data exchanged between two persons. For instance, an intruder could change health, police or banking records for his/her own benefit ,data theft is another main source of concern for security professionals, many organizations rely heavily on computer network systems and protecting personal information has become critical ,sabotage usually comes from an inside intruder. Often, the damage is caused by an employee and directed to hardware equipment, Sabotage usually involves an employee of the organization who tries to disrupt either the network system or components, such as network equipment, servers and so on. The reasons for sabotage are multiple, although greed and injustice are the more common, social engineering is becoming a common type of threat. It usually involves impersonation of an authorized body in order to retrieve login, Malicious software includes worms, viruses, and Trojans and logic bombs and can have diverse sorts of devastating effects this type of software often exploits operating system weaknesses or software flaws.

2.3.2 Threat applications

Most common applications of computer system threats include Virus, Worms, Trojans, Denial-of-service attacks, and Scanning /Reconnaissance.

NIST, (2006), define viruses as pieces of malicious software, often attached to legitimate documents which require human intervention to be activated. These types of software or malware can have various actions, from simply showing ad windows to erasing or changing the content of files, user tend to keep the spread of viruses going on by sharing files or sending emails.

R. C. Newman, (2006), considered Worms as a sub-class of viruses but differ from them because they can replicate themselves and spread across a network without any human intervention. They can have destructive effects on a host system, but can also affect networks by replicating and sending multiple copies of themselves across a network thus creating a DoS such as with the “Witty” worm. Trojans are also malicious pieces of software attached to legitimate files. Where they differ from viruses is that they are usually attached to applications which seem useful, but Trojans add hidden functions, such as remote shell access, thus compromising the security of the host. Such functions can enable remote access for an external intruder, hence creating a backdoor, or sending Valuable data outside the organization.

According to R. C. Newman (2006) DoS attacks aim at depriving legitimate users from access to resource. Such resources could be a server, a printer or any other type of device providing a service. There are three basic types of DoS consumption of limited resource, destruction/modification of configuration files and physical modification of network infrastructure. Attention will be drawn to the first type of DoS since it are more closely related to network traffic. DoS is the most common tool used by cyber terrorists in order to blackmail and take down businesses servers in exchange for a ransom

Reconnaissance tools and scanners are used in the first step of an attack, in order to gain information about the target.

According to K. Buzzard, (1999), different types of information can be gathered which these tools, such as open port numbers, network addressing scheme and topology, web applications security.

2.3.3 Threat consequences

Organizations which are victims of computer systems security breaches can experience loss or degradation mainly in three specific domains: performance, public image and monetary. The first domain is computer systems related and technical, whereas the other two are not.

K. Lan, A. Hussain, and D. Dutta (2003), shows that performance can be heavily affected by computer systems attacks. Through their experiment, they show that a DDoS attack can increase the mean latency for DNS lookup by 230% and the web latency by 30%. They also measure the spread of the Slapper worm in a simulation, and state that if the entire infected host would launch a coordinated DDoS, a network could be taken offline in a matter of minutes. Worms, viruses and Trojans can have different effects, depending on the designer's intentions.

Labib, (2004), shows this with a bank from which credit card number and accounts information were stolen; such an intrusion can considerably diminish the security credibility of a bank. Finally, the financial loss possible following a security breach is the domain which drives computer systems security.

2.4 Firewalls

In order to maintain security in an organization, IT professionals employ a diverse range of security tools. Among the most common are firewalls, IDSs and other host security oriented tools, such as anti viruses. This section highlights firewalls as well as their respective strengths and weaknesses.

M. J. Edwards, (1998), argues that the most common computer networks security tools consist of firewalls. They can be found in most of all corporate networks and form the first barrier against intrusions.

K. Ingham and S. Forrest, (2002), define a firewall as dedicated system or software application that inspects traffic against a set of rules. Without good configuration, firewalls are useless. Unfortunately, as well as being very popular they are also often misconfigured, allowing any traffic by default rather than denying all of it.

Ingham and Forrest (2002), goes ahead to define of firewalls as a device or group of devices which separates corporation assets from potentially dangerous external

environments, which could be the Internet, for instance. They also give a few criteria that have to be fulfilled in order to class a device as a firewall:

- i) A firewall should be at the boundary of two networks
- ii) All traffic entering or leaving the intranet should cross the firewall
- iii) A firewall should have the capability of allowing and dropping traffic, in order to enforce a security policy
- iv) A firewall should be resistant to direct attacks and have no direct user access
- v) Firewalls also offer the possibility to create a Demilitarized zone (DMZ), it is possible to place machines which offer services to the Internet in a DMZ .by doing so, the rest of the network remains protected in case of a breach in.

M.J Edwards (1998).defines the different types of firewall techniques exist and can be classified as packet filters, application layer gateway or stateful packet filters

2.4.1 Packet filters

According to M.J. Edwards, (1998), packet filters are the most popular and the simplest form of firewalls. They analyze traffic owing through and allow or drop traffic based on network and transport layers packet information such as source and destination Internet Protocol (IP) addresses and ports, direction, type of packets, and so on. This type of firewall is often implemented on edge routers with Access Control List (ACL) s applied on ingoing and outgoing interfaces.

According to H.Bidgoli, (2005), if a FTP session is considered, a client will send a request to the destination port of a server. The server will then reply to the client. The first TCP/IP packet will have a destination port of 21 and the reply will have a source port of 21. This means that the packet filter should have two specific rules to enable FTP sessions to cross the firewall. The problem here is that an intruder with knowledge of the target system can easily forge fake FTP packets that the firewall will allow, thinking that they are legitimate FTP responses.

2.4.2 Application layer gateway

K.Ingham and Forest, (2002), application layer gateways act as a relay for connections between inside machines and extranets. It sits between users inside and servers outside. A user who tries to connect to an Internet server will actually connect to the gateway, which will in turn carry out the request on behalf of the user to the external server. The server replies to the gateway, which will then forward the reply back to the user. For external users, a network using an application layer gateway appears as a single machine.

E.Roop (2007) indicates the advantages of gateways are that they can filter based on packets content, include a user level authentication and hide the structure of the network from external potential intruders.

K.Ingham and Forest (2002), show that the main drawbacks of application layer gateways are that not all services have usable proxies already existing and that they are relatively slow to process packets.

2.4.3 Stateful packet filters

According to M.J Edwards, (1998), these systems are a refinement of the packet filtering technology. It acts as traditional packet filters but also monitors connections for increased security. Stateful packet filters operate at the network and application layers of the Open Systems Interconnection (OSI) model. Stateful firewalls monitoring initial connections and allows replies to cross the firewall until the connection is close.

K.Ingham and Forest (2002).A similar method can be used for ICMP and UDP packets, although these protocols are not connection oriented. In most cases, different types of firewalls are combined together in order to increase the overall perimeter security.

Application layer gateways or stateful filters are often used as primary firewall and traditional packet filters are added after the firewall to avoid inexistent security in case of main firewall

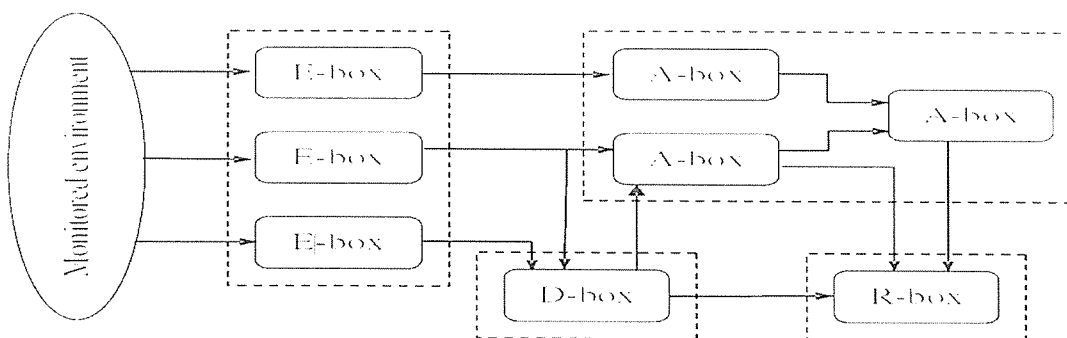
According to H.Bidgoli (2005), if a FTP session is considered, a client will send a request to the destination port of a server. The server will then reply to the client. The first TCP/IP packet will have a destination port of 21 and the reply will have a source port of 21. This means that the packet filter should have two specific rules to enable FTP sessions to cross the firewall. The problem here is that an intruder with knowledge of the target system can easily forge fake FTP packets that the firewall will allow, thinking that they are legitimate FTP responses.

2.5 Intrusion detection system

2.5.1 Definition

According to R.Bace and P.Mell, (2001), intrusion detection refers to the monitoring of events and the analysis for signs of intrusions. Intrusion detection systems (IDSs) are software applications which automate these monitoring and analysis processes. IDSs are typically used to detect attacks or violations not detected by other security means; to detect reconnaissance attempts preceding attacks such as with probes and scans. They can also be used to control the quality of an existing security design and administration, or to help diagnosis, recovery and correction of breaches in case of a current attack occurred

Figure 1: 2.1- IDS architecture



Source: P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez. (2008)

According P. Garcia-Teodoro, (2008), IDS captures monitored data through its sensors (“E-box”). It then compares this data in the analysis module (“A-box”) and stores it (“D-box”). It can finally react to a detected intrusion via a reaction component (“R-box”)

IDSs can also be sub-classified into many different categories, but one of the main differences includes the following two categories: host-based IDSs and network-based IDSs.

2.5.2 Host-based IDS and Network-based IDS

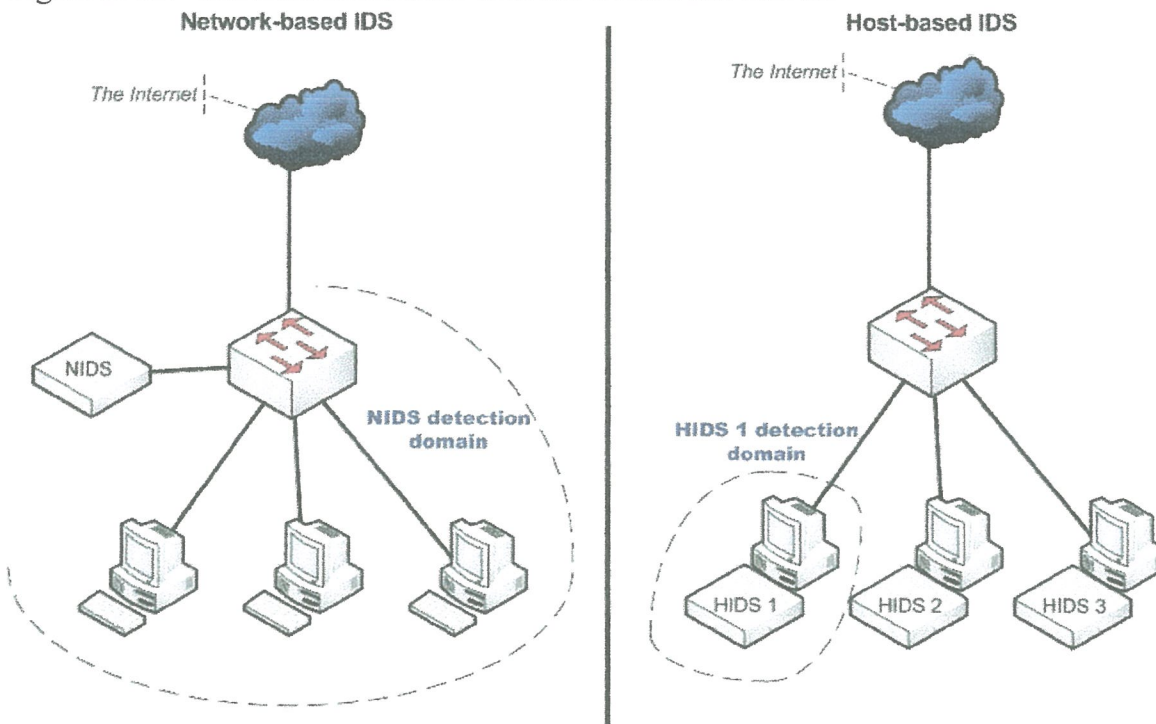
According to P.Garcia-Teodoro, J.Diaz-Verdejo, (2008), host-based Intrusion Detection System (HIDS) were the first type of intrusion detection systems to appear they are typically installed on the host they are monitoring and have access to the operating system information.

K.Ingham and Forest ,(2002), shows that HIDSs prove useful because they can detect encrypted attacks, by checking traffic before being sent or just received, and also because they can detect attacks targeted to the specific system and undetectable in network traffic, such as Trojans. Another advantage of HIDSs is that they can access system information, generating more accurate alerts and more detailed log files. Disadvantages include that they can monitor the single host they are running on, and have to be specifically set up for each host. Scalability is the main issue for HIDSs. They also use resources on the target host

According to H. Debar, M. Dacier, (1999), NIDS can monitor a segment of network to a large section of a network, depending on their placement. They function in promiscuous mode in order to capture network packets, thus they have very little impact on the overall network performances. Unfortunately they have a few disadvantages, including the fact that they cannot process encrypted packets and require the use of SPAN ports if attached to a switch in order to monitor all traffic going through the switch

J. Sommers, V. Yegneswaran, and P. Barford, (2004), indicates that the other main disadvantages of NIDSes is that they can have difficulties processing large amount of network packets if they are set up to monitor a large and/or busy section of the network

Figure 2: 2.2- The main differences between HIDSs and NIDSs:



Source: H. Debar, M. Dacier, (1999)

H. Debar, M. Dacier, (1999) also show how it is possible to measure the efficiency of an IDS. IDSs are often evaluated in terms of accuracy, performance and completeness. Accuracy is the ability of the IDS to flag as intrusive only packets that are part of an attack. The performance of an IDS is the rate at which events are processed, thus a good performance measure makes real time detection possible. Finally, the completeness of a system is the ability of detecting all the attacks that occurred in a given time. This measure is often the hardest to establish in a live environment because it is impossible to know exactly how many attacks were carried out and at which time.

Arguably, if such was possible, then there would be no need for IDSs. There are some key concepts related to IDSs: false-positives, false-negatives and true positives.

Table 1: 2.1-below provides definitions for these Alarm types.

Alarm Type	Definition
True-positive	IDS rightfully flags an attack as such
False-positive	IDS triggers an alarm although no attack is actually happening
False-positive	Real attack that the IDS does not flag as intrusion
True-positive	IDS does not flag legitimate events as attacks (most common situation)

Source: H. Zhengbing, L. Zhitang, and W. Junqi, (2008)

2.5.3 Anomaly detection

Anomaly detection technique applied to computer systems was discovered in 1986 when Denning proposed the idea that it could be possible to identify abnormal unusual behavior (anomalies) by comparing current behavior to a known normal state this statement was based on the assumption that attacks are clearly different from normal traffic. This “normal traffic” states are recorded in profiles. These profiles can either be generated via offline learning or the system can learn by analysis traffic in an online way Anomaly detection systems prove useful at detecting insiders’ attacks, as well as previously unknown attacks, known as “zero day”.

According to F. Gong, (2003), anomaly based IDSs are useful when it comes to detecting new threats, or different versions of known threats. Where signature-based IDSs prove very useful for detecting known attacks, it has been proved that evading such security systems can be accomplished relatively easily. Unfortunately, these advantages do not come without intrinsic drawbacks; the system must go through a training phase before any intrusion detection in order to build profiles for normal traffic.

According to W. Lee and S. J. Stolfo, (1998), anomaly detection IDSs rely on several methodologies.

Statistical based anomaly detection is one of the “simplest” and oldest methods, modelling statistics from different parameters. For example, Statistical Packet Anomaly

Detection Engine (SPADE) uses the time series model form of statistical approach, using timers, counters and order of arrival of events.

The other anomaly detection approaches involve different more or less complicated methodologies.

.As Gates and Taylor, (2007), state modern anomaly detectors are often based on Denning's assumptions which were valid at the time for HIDSs, but not anymore in the context of current networks and NIDSes. Such assumptions are for example that attacks are anomalous, that attacks are rare, that attack-free training data is available or that the false alarm rate should be under 1% to be acceptable. Gates and Taylor (2007) give an analysis which shows that all of the above assumptions can easily be challenged. Nowadays, attacks are more and more common with the increase use of the Internet, and intruders can manage to craft intrusive traffic like normal traffic. The attack-free training data remains one of the main issues with anomaly detection. Training a detector in a "live" environment might include attacks as normal behavior. However this issue is currently being actively researched.

2.5.4 Signature detection

According to H. Zhengbing, L. Zhitang, and W. Junqi, (2008), signature detection also called knowledge-based detection, is the most popular commercial type of IDSs. Signature detection systems use knowledge of known attacks, exploits and vulnerabilities and look for matching attacks patterns in network traffic or system events. The accuracy of such systems is considered to be very good because they tend to have a low rate of false positive alarms.

According to A. Patcha and J.-M Park, (2007), this type of systems can detect known attacks reliably as well as having a low false-positive rate, these systems produce detailed data about the attacks. Since the signature is known and detected, the attack is clearly recognizable, making the network administrator's work easier. In order to keep a good

completeness standard, the signatures database has to be maintained up to date very frequently.

K. Ingham and S. Forrest, (2002), the main drawbacks of signature detection are that signatures can be easily escaped with morphs of known attacks and that these systems can only detect attacks related to their knowledge database.

Debar et al (1999,) shows that different methodologies can be used to achieve the same misuse detection goal. Among these methods are expert systems, signature analysis, Petri nets or state-transition analysis. The most commonly applied to commercial IDSs is the signature analysis method, which reduces patterns of attacks to the lowest level of semantics.

According to N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, (2005) examples of well-known signature detection IDSs include Snort (open source tool) and Bro (commercial tool)

2.5.5 Hybrid systems

A. Patcha and J.-M. Park, (2008), shows that the detection capabilities of IDSs can be improved by taking a hybrid approach, taking the best of both signature and anomaly detection.

2.6 Other security Tools

2.6.1 IDS related tools

One of the main issues with anomaly detection IDSs is the large amount of alarms reported, mainly false-positives. Such problem can affect the judgment of the system administrator who has to process all these alarms and might miss the attacks among the load of logs.

J. B. Colombe and G. Stephens,(2004), presents a method aiming at grouping alarms generated by IDSs into clusters, which have the same root cause or source attack. Solving the problems generating these alarms, which are typically false-negatives, helps the network administrator do his job more efficiently. The benefits of this method are that the alarm bulk is reduced by 90%, leaving the human analyse fewer alarms.

2.6.2 Antivirus

According to R. Lippmann, S. Webster, and D. Stetson, (2002), antivirus are very common in any organization. Installed on each machine of a network, they provide local defense against a wide range of malware (worms, Trojans, viruses, root-kits, etc.). To do so, they operate in a similar way as signature based IDSs do: they scan the host system for matching patterns of threats with a database of threats signatures.

2.7 Testing of Intrusion detection systems

This part focuses on IDS s evaluation as they constitute the main topic of this research. The main challenge in IDSs deployment is assessing and comparing performances of their systems with other IDSs (H. Bidgoli, 2005). These evaluations are needed and driven by the fact that security systems have to prove what they are capable of detecting, and how well they operate compared to the each other. Also, Woloch, (2006) states that testing of intrusion detection systems is not as advanced as one would hope. This section thus presents the different current methods of intrusion detection evaluation and testing and their characteristics.

2.7.1 Evaluation metrics for IDSs

K. Labib, (2004), mentions detection rate and false alarm rate as the best suited metrics. The detection rate is equivalent to the number of intrusions detected divided by total intrusions injected in the traffic. The false alarm rate is equivalent to the false-positive rate of the IDS (as seen in section 2.4, a false-positive occurs when the IDS flags legitimate traffic as intrusive or abnormal).

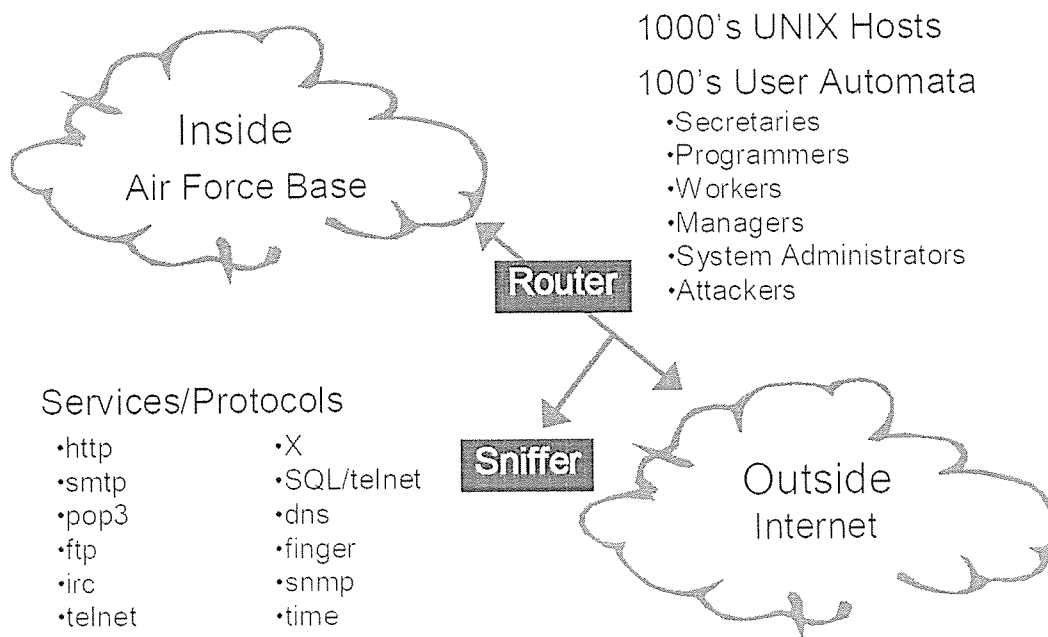
Sommers et al, (2004), use efficiency, effectiveness, packet loss and Central Processing Unit (CPU) utilisation as metrics. The first two metrics are equivalent to the two rates presented above, whereas CPU utilisation and packet loss are new measures, useful to determine how a system copes under traffic load.

Graves et al, (2005), emphasises on the necessity of this latter packet loss measure. The efficiency of a system is in fact the false-positive alarms occurrence ($\text{Efficiency} = \frac{\text{True positive}}{\text{All alarms}}$). The closer to 1 it is, the better positives all alarms the system can flag real attacks only. The effectiveness produces the false-True=negative alarm rate of the IDS (Effectiveness). This metric shows positives all positives the events missed by the IDS. It has to be noted that these metrics apply to any type of IDS

2.7.2 Offline Evaluation

According to J. McHugh, (2000), offline evaluation consists of recreating datasets of network traffic including attacks without recreating the whole network topology. The use of tcpdumps and replay tools allow such type of evaluation. The most commonly used datasets were created by Defense Advanced Research Projects Agency (DARPA) /Massachusetts Institute of Technology (MIT) Lincoln Labs in 1998 and 1999, called 1998 DARPA set and 1999 DARPA set, and also sometimes called Intrusion Detection Evaluation (IDEVAL) datasets. The DARPA sets are simulations of network traffic based on observation of real network traffic including common attacks, which aim at providing blind evaluation material for researchers. These datasets were captured at the edge of a network, at the border router. Figure 2.2 presents the structure and services characteristics used in the DARPA datasets network

Figure 3: 2.3- Darpa experiment setup



Source: D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. Mcclung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman,(1998) The 1998 darpa off-line intrusion detection evaluation

According to D.J.Fred,I.Graf,J.W.Haines (1998), the 1998 DARPA set includes 7 weeks of training data with labelled test data and 2 weeks of unlabelled test data. During the first test competition, 8 IDSs were tested. The data set includes also over 300 instances of 38 attacks. The 1999 DARPA set presents over 5 million connections over 5 weeks: 2 were attack-free and 3 weeks included attacks. Another data set was created in 1999, based on the 1998 DARPA set: the 1999 Knowledge Discovery and Data mining (KDD) Cup, created for a machine learning evaluation competition

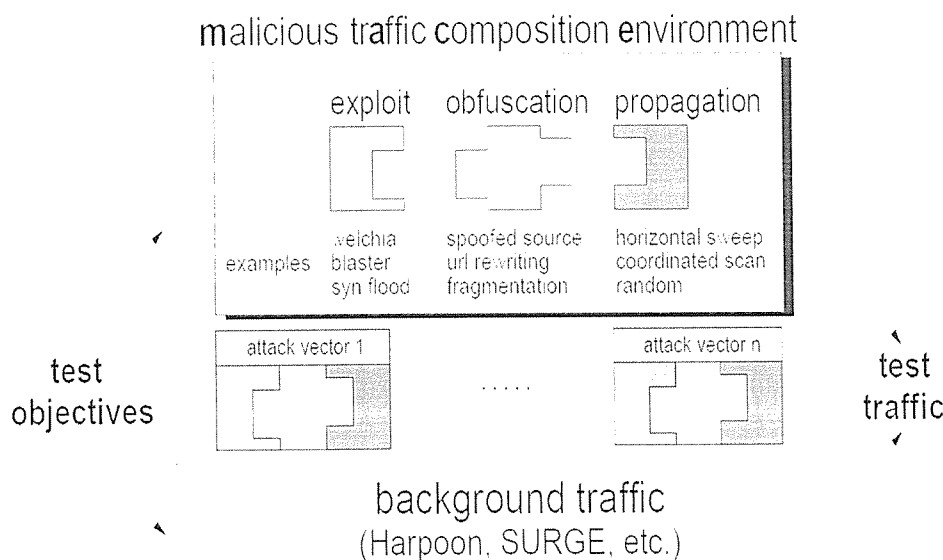
2.7.3 Online Evaluation

After seeing the shortcomings of current offline evaluation, there is a critical need for realistic traffic and attack generators, as well as data sets mixing both type of traffic in a realistic manner. Current researchers focus their work on simulation testbeds and attacks

generators. Lincoln Labs' work aiming at creating an online testbed resulted in the Lincoln Adaptable Real-time Information Assurance Testbed (LARIAT) tool

P. Fogla and W. Lee, (2006), LARIAT is capable of generating realistic background user traffic and real network attacks. It was created to overcome the issues inherent to the DARPA sets, in order to create a next generation of testbed. The main two goals of LARIAT are supporting real-time evaluation and creating easily deployable and configurable testbed. It simulates internal and external networks, it is thus possible to evaluate IDSs "plugged" in between both simulated networks. Another two tools, which used together achieve similar goals as LARIAT are Malicious traffic Composition Environment (MACE) and Harpoon, both developed by Sommers et al. figure 2.3 shows how these two tools can be used to achieve LARIAT's goals

Figure 4: 2.4-Trident framework



Source: J. Sommers, V. Yegneswaran, and P. Barford (2004)

According to J.sommer, H.kim, and P.Barford, (2004), harpoon is a low-level traffic generator, used to create benign realistic traffic based on real network packets traces it allows to modulate a mixture of benign and malicious traffic in a realistic way, as well as controlling the temporal arrival of each type. MACE is a performance benchmarking tool

and malicious traffic generator. Sommers et al. released a new tool called Trident, which includes MACE and Harpoon as well as extra novel features such as DARPA attacks recreation for instance.

G. Vigna, W. Robertson, and D. Balzarotti, (2004), presents an evaluation of two open source signature based IDSs (Snort and ISS Secure) with a framework generating mutant exploits. This tool is “an automated mechanism to generate functional variations of exploits by applying mutant operators to exploit templates”.

G. Vigna, W. Robertson, and D. Balzarotti (2004), carried out the evaluation that uses ten common exploits, including DoS attacks, buffer over flows, targeted to different operating systems, including OpenBSD, Linux distributions and Windows OS, and different common services such as FTP,Hyper Text Transfer Protocol (HTTP) and Secure Sockets Layer (SSL)). It shows that 10 out of 10 basic exploits were detected, against 1 out of 10 for mutated exploits. This shows that it is relatively easy to evade signature detection by using mutant exploits.

P. Fogla and W. Lee (2006) provide similar framework, Polymorphic Blending Attacks (PBA). In their paper they provide a formal framework for creation of mutants, like the previous tool. It tests the efficiency of this tool against an anomaly based IDS. The outcome of this evaluation shows that it is also relatively easy to evade anomaly detection by using morphs of attacks.

Finally, J. Sommers, V. Yegneswaran, and P. Barford, (2006), presents a framework for defining test cases scenarios. The authors prove that the existing classification of possible attacks does not match all the needs of IDS evaluation and testing. Thus, they provide a framework covering all the characteristics of attacks in order to create a complete scenario evaluation. , conclude that research should still be done in order to match real attacks to each category. Evaluation of IDSs is as much of a challenge as designing efficient algorithms for intrusion detection.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter provides the frame work within which data was collected and presented. It covers the research design and details the basic design, as well as the main goals of the experiment carried out as part of this dissertation. As seen in Section 2.7, there are some limitations to existing literature concerning IDS evaluation. First of all, there is a need for testing directly two different types of NIDS, in other words anomaly and signature detection. This Chapter presents experiments which attempt to take into consideration both of these needs. Section 4.2 shows to what extent this experiment achieves these goals. The initial objective of this experiment was to set up a testbed for two different types of NIDS and generate simulated background traffic as well as range of exploits. Such an experiment proved too generic since the choice of exploits ready to use was relatively small compared to the amount of existing exploits. Instead, the experiment was split in two: a first experiment on the learning window variation of an anomaly IDS, and a second experiment testing two different types of IDS in a specific, well-defined scenario. S.peisert and M.Bishop (2007), states that a valid computer security experiments should consist of only one varying component. The experiments carried out in this paper meet these criteria. The following sections define an overview of the testbeds used.

3.1 Research design

The purpose of this study was to analyze and evaluate network intusion detection system by comparing anomaly detection method and signature detection method

3.2 Target Population

The research focused on system/network administrators maintaining various network systems and some user of the these systems

3.3 Research instruments

For research precision purposes, three instruments were used in the data collection process. This approach was taken owing to the fact that no single method of collecting data is 100% accurate. To enable logical conclusions to be made out of the research findings, the instruments that were employed in collecting data from various respondents included; interviews, observation schedules and literature survey.

(a) Interviews

Interview sessions with the systems administrators and users from various organizations were carried out. The interviews encouraged lots of individual participation and acted as an effective tool in gathering insights into the state of the intrusion detection methods or system as well as providing solutions to the problem areas. Considering that the only way of getting the right answer is by addressing the right questions to the right people, I drafted an interview guide that contained structured questions which acted as a tool to organize my thoughts and served as a fallback position incase lost track of events as a result of being so engrossed in the interview.

. A sample of the interview guide is available at Appendix A.

(b) Observation

Observation involves noting something and giving it significance by relating it to something else noticed or already known. Direct observation which involved looking at the security configurations on the host machines as well as servers.

This enabled to understand the existing security configurations of the network resources that are in place.

(c) Literature survey

This involved reading published materials (both electronic and printed) concerning the study. This determined what documentation had already been completed on the subject and thus gains a better understanding of the many facts of the problem at hand.

3.4 Data presentation/analysis

The collected data was analyzed and processed into meaningful and relevant information. It was accorded percentages to facilitate analysis. Qualitative data was analyzed by

comparison to findings already known and conclusions were made depending on how the findings related to the research questions.

3.5 Logical network architecture

The basic network architecture is composed of a switch, server computer and clients. The background traffic is split into two IP address ranges. A switch is used exchange data among the nodes.

3.6 Training window experiment

This experiment aims at demonstrating any effects that a variation of training window length could have on an anomaly-based IDS. Figure 3.1 presents a high level diagram of the tested setup for this experiment. The DUT is represented by the station running the IDS. This station is linked to a switch and monitors all network traffic crossing this network device. The traffic generator is used to produce benign background traffic for anomaly system profile creation. Ideally, this station should produce this type of traffic with a traffic generation simulation tool such as Harpoon for instance. The exploit generator is used after the profile generation phase has been completed. How well the IDS detect the exploit generated will help compare each different learning window and allow extracting conclusions from these observations. The anomaly-based IDS will be subjected to different learning periods. For each period, the profile created will be stored for the next experimental phase, being the attack detection. After this profile generation phase, the IDS will be subjected to a mix of benign background traffic and malicious traffic. The amount of malicious traffic injected is known, thus the different types of alarms shown in Table 2.1 can be known, and measures like effectiveness and efficiency shown in Section 2.7.1 can be evaluated.

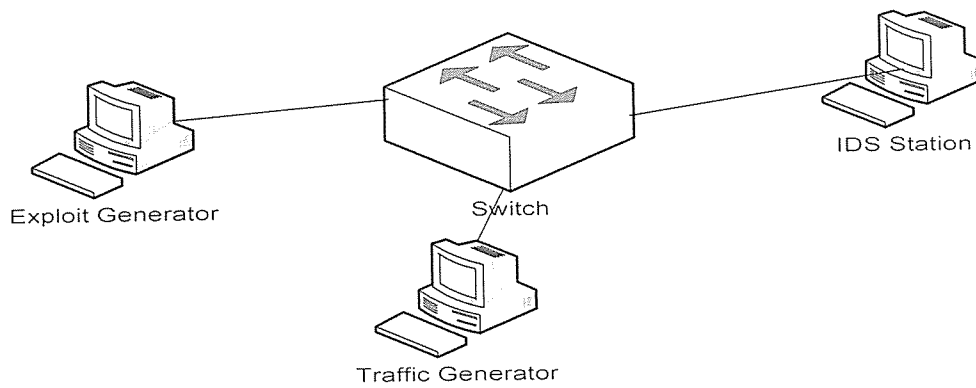


Figure 5: 3.1 -Training window experiment design

3.7 Scenario

This experimental scenario is realised in order to focus this research on a specific type of threat rather than only available threats. For this scenario, the following background is to be considered. A renowned microfinance branch computer networks system includes an FTP server hosting highly sensitive data, such as customer account details for example. Currently, the firm uses the following tools as part of its security system: a firewall at the boundary with the untrusted network, antivirus on local machines and built-in IDS on the gateway router analyzing traffic going in and out of the trusted network, such as on a Cisco router. Figure 3.2 shows a high level representation of such a computer system. The security at the boundary of the firm network is optimum, but the IT network administrator is worried about threats present on the inside of their network. Insiders threats are multiple (Section 2.3.1), although here the main concern is data theft from the FTP server, only protected by a username and password combination. In order to protect the branch from such a threat, the security staff would like to know which type of NIDS would be best suited in this case the boundary of the firm network is optimum.

There are some considerations to take into account with regards to this scenario. Sensitive data would probably not be stored on a simple FTP server in a real case environment, and the access to such a server would probably be more securely controlled. The simplistic approach used in this scenario is chosen due to time considerations and testing focus: a FTP is faster to breach than a more secure server, and this experiment is focused on NIDS rather than server security. This scenario can show which IDS is best for such an

environment. Adding additional security measures could not do any harm but make the whole system more secure.

Figure 3.2 shows how this scenario is translated into a usable testbed for NIDS evaluation. The testbed is very similar to the one used in the learning window experiment, with the difference of an extra machine running an FTP server. The scenario threat is data theft. Data theft is usually composed of a collection of exploits following the steps highlighted in Section 2.3.1

. In this case, the data theft consists of

- i) Live IP scan
- ii) Ports scan on live address
- iii) Brute force attack on FTP username/password
- iv) Data theft

The exploit generation station will carry out every step of a data theft threat

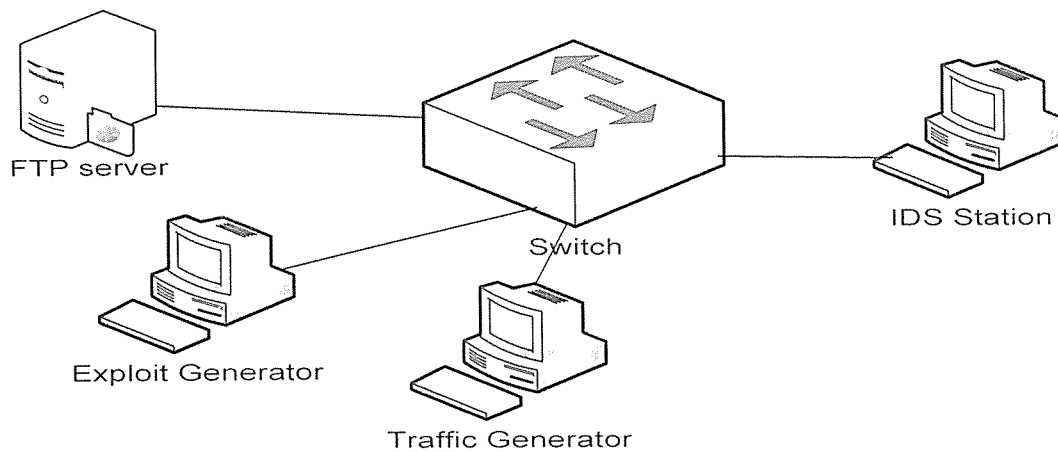


Figure 6: 3.2- Scenario experiment design

3.8 Hypothetical results

It is possible to draw hypothetical results from the normal behavior of IDSs and exploits generated. With regards to the training window experiment, the anomaly-based IDS should flag all packets which are part of the attack generation if they are very different from background traffic, and the same goes for all different learning window length. Any new packet different from the traffic seen by the IDS should be flagged anomalous. This

hypothesis depends on the method used by the anomaly-based IDS to create normal traffic profile, and as above-mentioned, is dependent on the traffic similarity. The signature-based IDS should flag the steps for which it has signatures. Portscans, IP address scans and FTP attacks detection are commonly implemented in such IDSs. On the other hand, the signature-based IDS cannot detect the data theft since this is considered normal traffic from a signature point-of-view (Section 2.5.3). With regards to the anomaly detection, the anomaly-based IDS should flag any packet that is very different from the profile as anomalous. Thus, it should flag any new scan try, repetitive fast FTP connections (brute force attack) and unusual data transfer to odd stations as anomalous. The experimental designs demonstrated in this chapter are in their simplest forms and allow a reliable performance analysis of different types of IDS. The first experiment will explore the impact of different learning window lengths on anomaly detection, while the scenario experiment will determine which type of IDS is best suited to uncover data theft.

CHAPTER FOUR

PHYSICAL NETWORK ARCTECTURE

4.1 Introduction

This chapter implements the experimental designs presented in Chapter three and describes the tools used for each experiment, as well as the procedure to install and configure each of them.

4.2 Background traffic generation

As Section 3.1 describes, the experiment should have a tool producing simulated realistic background traffic. DARPA data set was used in order to create background traffic. This experiment recreates the DARPA set as live traffic through a network composed of a couple of network devices. To achieve this, the traffic traces have to be prepared prior to being sent across the network.

4.3 Intrusion detection system

This section shows which IDSs which where chosen to be tested. With regards to the anomaly based IDS, the choice was very limited. Some of the IDSs

4.3.1 Signature detection

The signature-based IDS chosen for both experiments is Snort. This IDS was chosen since it is free, extremely powerful and widely used by researchers . Since signature detectors are only as good as the signatures they use, Snort uses the “Sourcefire VRT Certified Rules” version 2.4 for unregistered users. This set of rules contains a large number of signatures used to detect diverse threats, such as DoS attacks, worms and viruses, web server’s attacks.

The following can also be set:

i). WinPcap (Windows Packet Capture Library) is a packet-capture driver. Functionally, this means that WinPcap grabs packets from the network wire and pitches them to Snort.

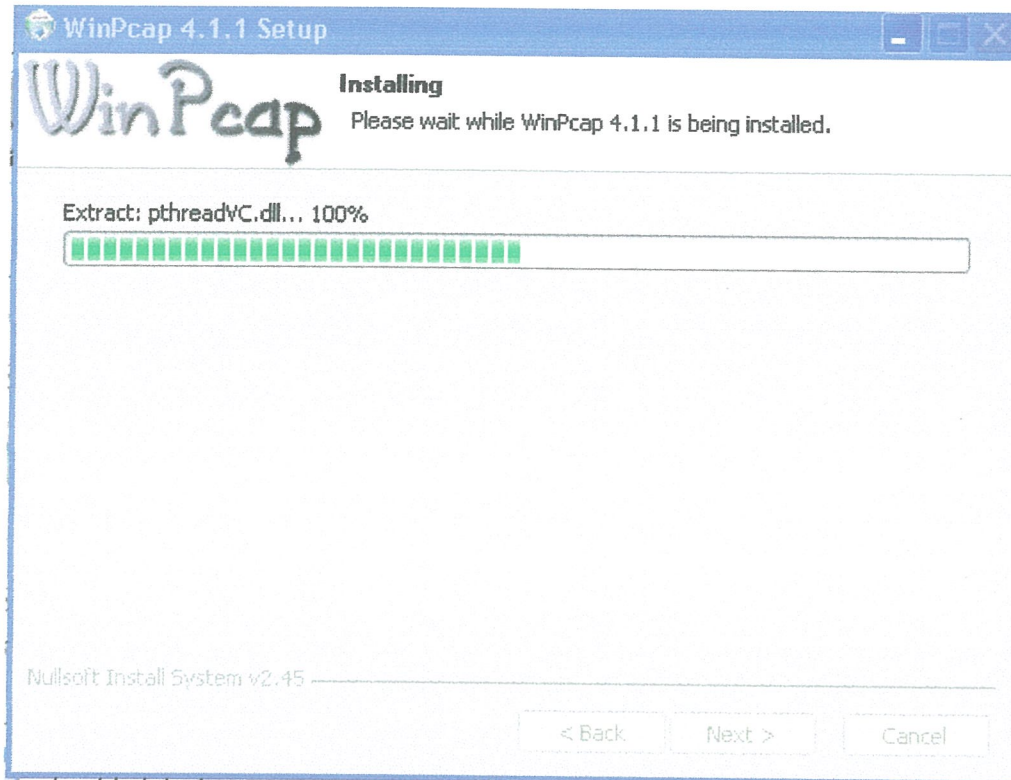


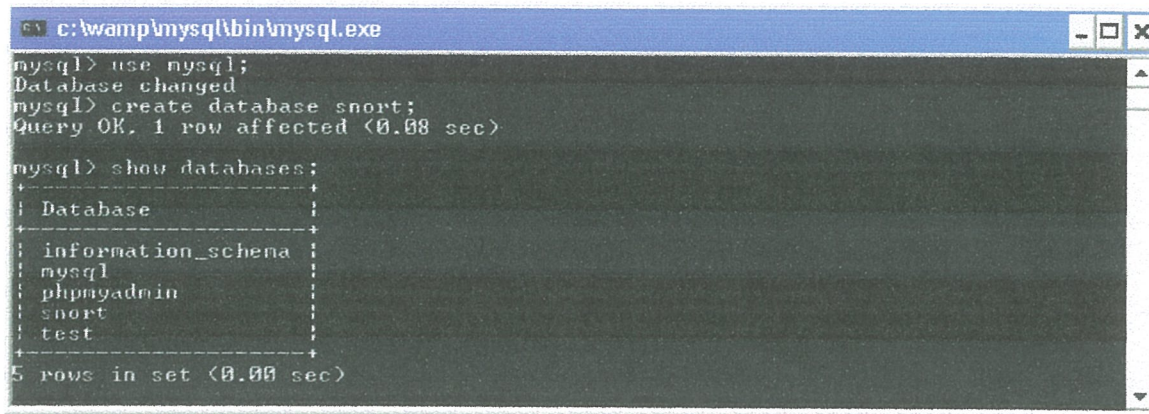
Figure 7: 4.1 WinPcap installation.

2. Mysql Snort has to send alerts to the MySQL database. This has therefore to be configured.

Configuring mysql with snort

- a) Create the Snort databases

C:\Snort\etc



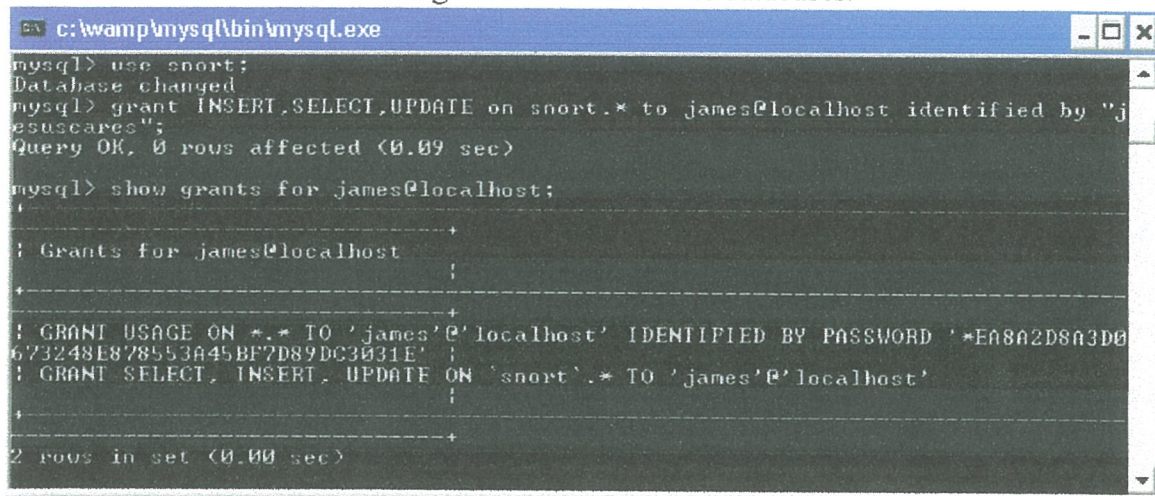
```
c:\wamp\mysql\bin\mysql.exe
mysql> use mysql;
Database changed
mysql> create database snort;
Query OK, 1 row affected (0.08 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| phpmyadmin |
| snort |
| test |
+-----+
5 rows in set (0.00 sec)
```

Figure 8: 4.2-creating snort database

b) Creating Snort's user accounts

Snort use this account when it logs in to add data to its databases.



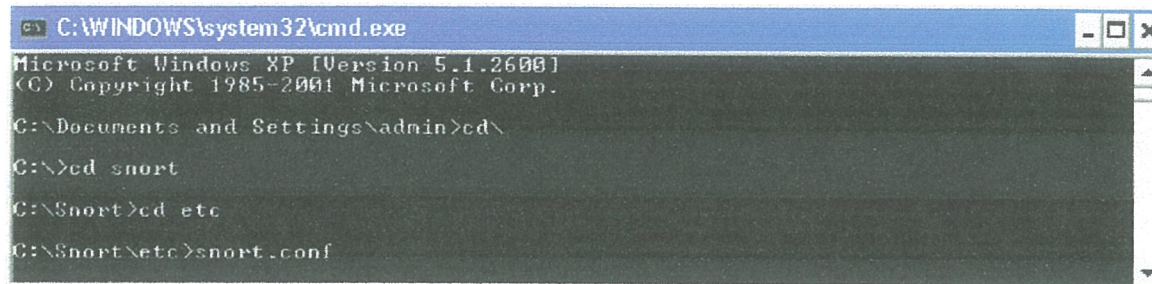
```
c:\wamp\mysql\bin\mysql.exe
mysql> use snort;
Database changed
mysql> grant INSERT,SELECT,UPDATE on snort.* to james@localhost identified by "jesuscared";
Query OK, 0 rows affected (0.09 sec)

mysql> show grants for james@localhost;
+-----+
| Grants for james@localhost |
+-----+
| GRANT USAGE ON *.* TO 'james'@'localhost' IDENTIFIED BY PASSWORD '*EA8A2D8A3D0673248E878553A45BF7D89DC3031E' |
| GRANT SELECT, INSERT, UPDATE ON 'snort'.* TO 'james'@'localhost' |
+-----+
2 rows in set (0.00 sec)
```

Figure 9: 4.3-creating snort user account

c) Snort installation

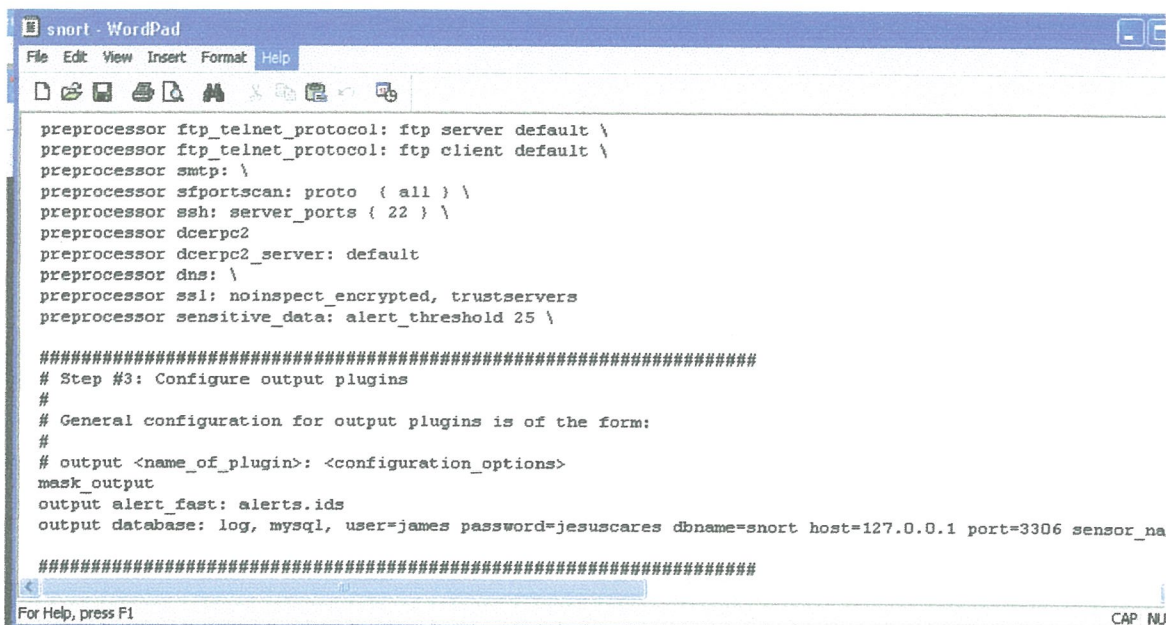
Double-click the executable installation file and the GNU Public License appears then follow the wizard. A new Snort installation requires a few configuration points. Conveniently, one file has all the configuration settings required (snort.config).



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd\
C:\>cd snort
C:\Snort>cd etc
C:\Snort\etc>snort.conf
```

The following configuration options in the snort.conf file are essential to a properly functioning Snort installation: Network settings, rules settings, output settings include settings. Figure below shows snort configuration file (snort.conf)



```
snort - WordPad
File Edit View Insert Format Help
preprocessor ftp_telnet_protocol: ftp server default \
preprocessor ftp_telnet_protocol: ftp client default \
preprocessor smtp: \
preprocessor sfportscan: proto { all } \
preprocessor ssh: server_ports { 22 } \
preprocessor dcerpc2
preprocessor dcerpc2_server: default
preprocessor dns: \
preprocessor ssl: noinspect_encrypted, trustservers
preprocessor sensitive_data: alert_threshold 25 \

#####
# Step #3: Configure output plugins
#
# General configuration for output plugins is of the form:
#
# output <name_of_plugin>: <configuration_options>
mask_output
output alert_fast: alerts.ids
output database: log, mysql, user=james password=jesuscares dbname=snort host=127.0.0.1 port=3306 sensor_na

#####
For Help, press F1
```

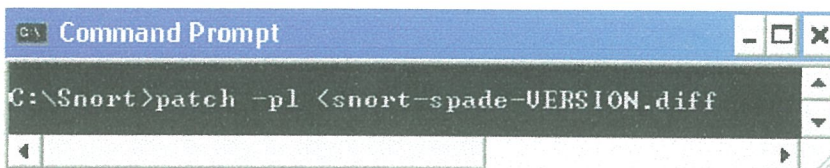
Figure 10: 4.4 snort configuration file

4.3.2 Anomaly detection

As mentioned in Section 4.3, open source anomaly detectors are not commonly found on the Internet. The only anomaly-based IDS found suiting the needs of this experiment is SPADE. SPADE is a preprocessor plug-in for Snort. It achieves anomaly detection by assigning anomaly scores to every packet analyzed. This anomaly score is based on the probability of the event, calculated following a combination of parameters such as source

and destination ports and IP addresses the first issue with SPADE is finding the source files because this project has been discontinued since 2003.

These files are available through the tool Open Source Security Information Management (OSSIM), in the “ossim/contrib/snort/” folder of this distribution. The second issue comes with the installation and the configuration of SPADE: the official documentation is inexistent, and general literature on it is rare. Most writers indicate that SPADE is integrated to all versions of Snort above version 1.7, but when this instruction are followed, SPADE source was found in the Snort install or on Snort website in the contribution section. The method used in this experiment to install SPADE was found in a post on the Snort forum. The first step is to copy the file with the .diff extension into the top directory of Snort. Then, run the following command and install



```
C:\Snort>patch -p1 <snort-spade-VERSION.diff
```

4.4 Training window experiment

As seen in Section 3.3, this experiment aims at testing the impact of different learning window lengths for anomaly-based IDSs.

4.4.1 Experimental parameters

The only experimental variation in this experiment is the training window length. Before launching any attack, the anomaly-based IDS are trained for 5 minutes, 10minutes and 30 minutes. Straight after the training period, the first attack is launched. There is then 3 minutes of only benign traffic before a second identical attack is launched. It has to be noted that background traffic is produced on a continuous basis throughout the experiment. This experiment last 4 minutes for each runs (30 seconds attack + 3 minutes + 30 seconds attack).

4.5 Scenario

As seen in section 3.4, this experiment aims at testing the detection difference between anomaly-based IDS and signature-based IDS.

4.5.1 Exploit generation

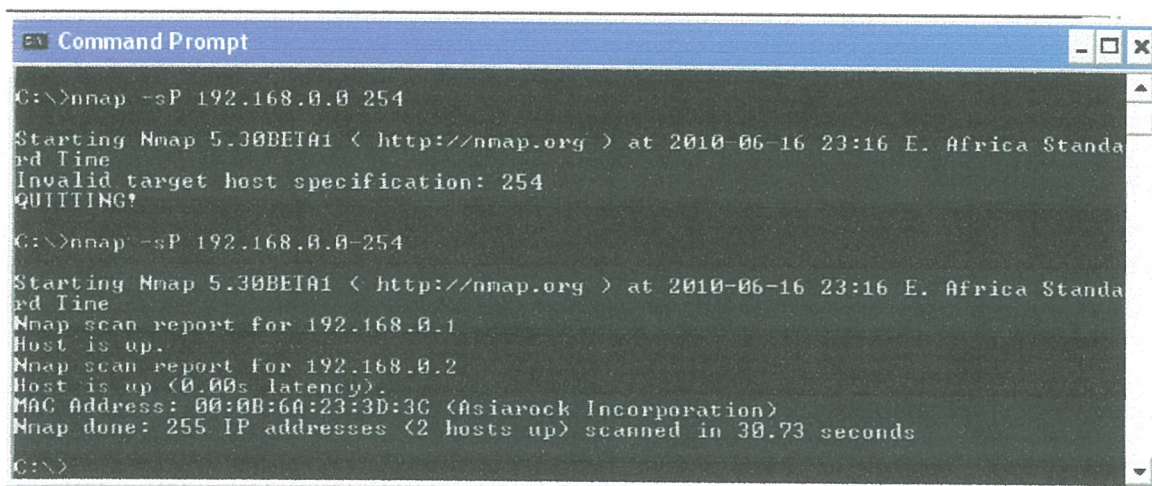
As shown in Section 3.4, the data theft scenario attack consists of the following exploits:

- i) Live IP addresses scan.
- ii) Portscan on live addresses.
- iii) Brute force attack on FTP username/password.
- iv) Data theft.

The first two scans are run with the Nmap v4.76 software. The command used to run the live IP addresses scan is:

nmap -sP 192.168.1.0-254

Figure 4.5 represents a screenshot of this reconnaissance step. It is possible to see that the IP address 192.168.0.1 and are up with their MAC address



```
Command Prompt
C:\>nmap -sP 192.168.0.0-254
Starting Nmap 5.30BETA1 ( http://nmap.org ) at 2010-06-16 23:16 E. Africa Standard Time
Invalid target host specification: 254
QUITTING!
C:\>nmap -sP 192.168.0.0-254
Starting Nmap 5.30BETA1 ( http://nmap.org ) at 2010-06-16 23:16 E. Africa Standard Time
Nmap scan report for 192.168.0.1
Host is up.
Nmap scan report for 192.168.0.2
Host is up (0.00s latency).
MAC Address: 00:0B:6A:23:3D:3C (Asiarock Incorporation)
Nmap done: 255 IP addresses (2 hosts up) scanned in 30.73 seconds
C:\>
```

Figure 11: 4.5 –Reconnaissance with Nmap step 1

The second reconnaissance scan, the portscan of the server is executed with the following command:

nmap -sS -sV -P 0 -T5 -O 192.168.1.1

Figure 4.4 shows the output of this command. It is possible to notify the TCP port 21 being open on the server, as well as other details about the OS. . The next step of this data

theft attack is using brute force in order to gain unauthorised access to the server password. In order to achieve this task, Hydra v5.4 for Microsoft Windows is used. Hydra is a fast network login cracker which supports FTP

```

C:\>nmap -sS -sU -PO -T5-0 192.168.0.2

Starting Nmap 5.30BETA1 < http://nmap.org > at 2010-06-17 00:19 E. Africa Standard time
Nmap scan report for 192.168.0.2
Host is up (0.00s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp
80/tcp    open  http         Apache httpd 2.0.58 ((Win32) PHP/5.1.4)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows XP microsoft-ds
443/tcp   open  ssl/http     CrushFTP httpd
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
2222/tcp  open  ssh          (protocol 2.0)
3306/tcp  open  mysql        MySQL (unauthorized)
8080/tcp  open  http         CrushFTP httpd
2 services unrecognized despite returning data. If you know the service/version,
please submit the following fingerprints at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF:Port21-TCPU=5.30BETA1%I=7%D=6/17%Time=4C194004%P=i686-pc-windows-windows%R(
NULL,37,"220-Welcome\x20to\x20CrushFTP4!\r\n220\x20CrushFTP\x20Server\x
SF:\x20Ready!\r\n")%R(GenericLines,37,"220-Welcome\x20to\x20CrushFTP4!\r\n2
SF:20\x20CrushFTP\x20Server\x20Ready!\r\n")%R(Help,5F,"220-Welcome\x20to\x
SF:20CrushFTP4!\r\n220\x20CrushFTP\x20Server\x20Ready!\r\n550\x20Command\x
SF:20not\x20recognized\x20or\x20allowed\.\r\n")%R(SMBProgNeg,37,"220-Welco
SF:me\x20to\x20CrushFTP4!\r\n220\x20CrushFTP\x20Server\x20Ready!\r\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF:Port2222-TCPU=5.30BETA1%I=7%D=6/17%Time=4C194004%P=i686-pc-windows-windows%R
(NULL,2B,"SSH-2.0-http://www.crushftp.com/\x20SERVER1\r\n");
MAC Address: 00:0B:6A:23:3D:3C (Asiarock Incorporation)
Service Info: Host: Welcome; OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.
org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 39.00 seconds

C:\>

```

Figure 12: 4.4 –Reconnaissance with Nmap step 2

The following command is used to launch Hydra:

```

C:\WINDOWS\system32\cmd.exe - hydra -l admin -P passlist.txt -t 5 192.168.0.2 ftp

C:\hydra 5.4 win>hydra -l admin -P passlist.txt -t 5 192.168.0.2 ftp

```

The passlist.txt file holds 256 passwords among which only one is valid. The “-t5” option limits Hydra to use only 5 concurrent threads rather than the 16 default. This limitation of Hydra was identified while testing the tool. With the default configuration, Hydra did not return the correct password. The same observation was made for any thread setting higher than 5. This technical difficulty slowed the password cracking process slightly, given the

fact that the password possibilities were relatively short. Indeed, with 16 threads, the password can be obtained in tenths of seconds, while with 5 threads, it takes

4.5.2 FTP server

The machine acting as the FTP server runs a dedicated FTP server software rather than using the built-in FTP server tool. CrushFTP v4.9.3 chosen because it offers a centralized method of management, from usernames and passwords to log files. It also offers many “live” details on the ongoing FTP sessions, such as the number of active connections, the number of failed and successful logins, and so on (Figure 4.5). The target file of the data theft on the FTP server is a file renamed to “Scenario” in order to mimic a large database. There are three considerations to take into account with the setup of the FTP server. First of all, as mentioned in Section 3.4, FTP is not a secure

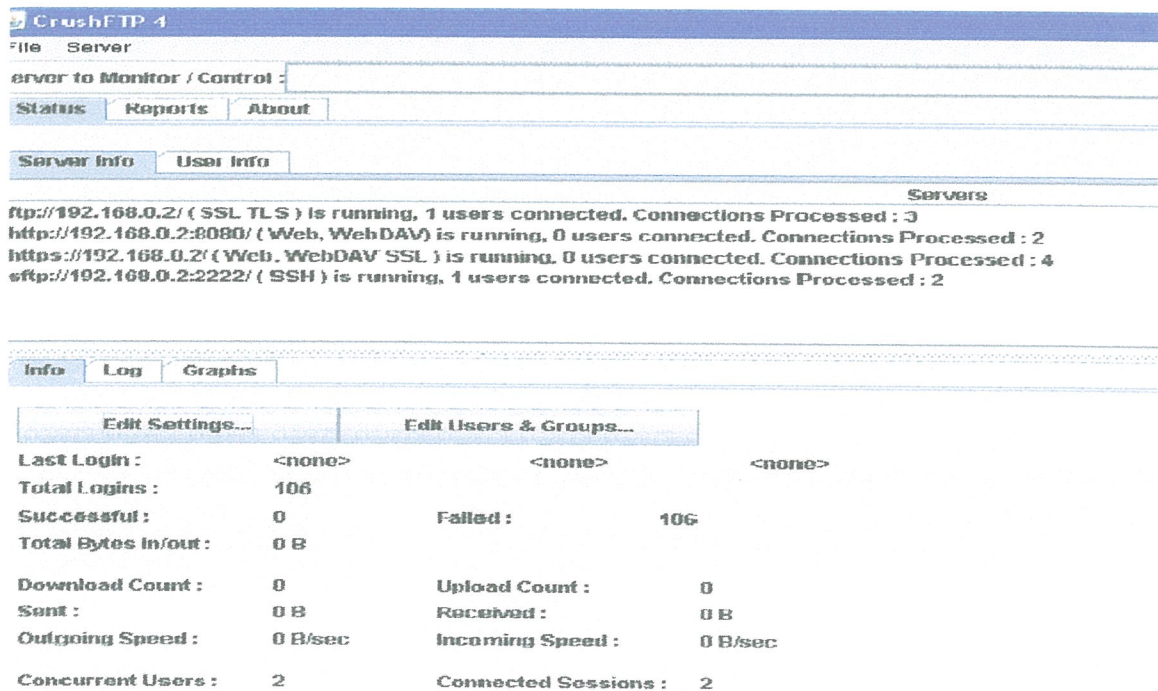


Figure 13: 4.7 ftp server

protocol. In a real environment, a more secure protocol such as Secure Shell (SSH) for instance would certainly be preferred. The second security weakness to take into account

is the weakness of the username/password combination used to log into the FTP server. This pair is set to “admin” as username and “tail” as password. The password is chosen among the list of all possible combinations of four characters (t, a, i and l). This means that there is 4 256 possible combinations, which is not highly secure. Finally, the FTP server is setup in a way that allows a password brute force attack. The settings limiting the rate of failed login attempts in a period of, time is set high on purpose in order to allow Hydra to operate at reasonable speed without getting blocked by the server security. These intentional security flaws were implemented to allow the experiment to fit in the given project timescale. With these parameters, it is possible to achieve data theft within five minutes. If SSH and server login limitation were implemented, it would take longer to achieve the same result. The second reason for these flaws to be present is the fact that this experiment aims at testing NIDSes rather than achieving the best server security possible.

4.5.3 Experimental parameters

The experimental parameters for the scenario experiment are as follow:

- a) FTP server username: admin.
- b) FTP server valid password: tail.
- c) Training window length: 10 minutes.
- d) Data theft length: 5 minutes

Table 2: 4.1 –Experimental tools summary

Tool	Version	operating system
snort	2.8.6	Window xp sp2
spade	2.3	Window xp sp2
Nmap	5	Windows xp sp2
Hydra	5.4	Windows xp sp2
Tcpdunp	3.1.1	Windows xp sp2
crushftp	4.0	windows xp sp2
wincap	4.1	Windows xp sp2
mysql	5.0	Windows xp sp2

Figure 14: 5.1–1st attack SPADE detection Rate

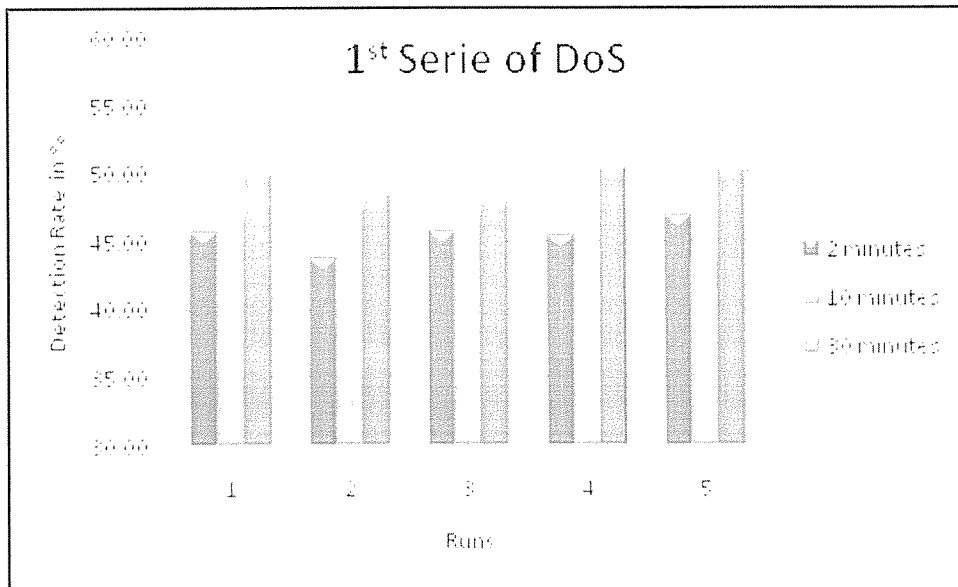


Figure 15: 5.2– 2nd attack SPADE detection rate

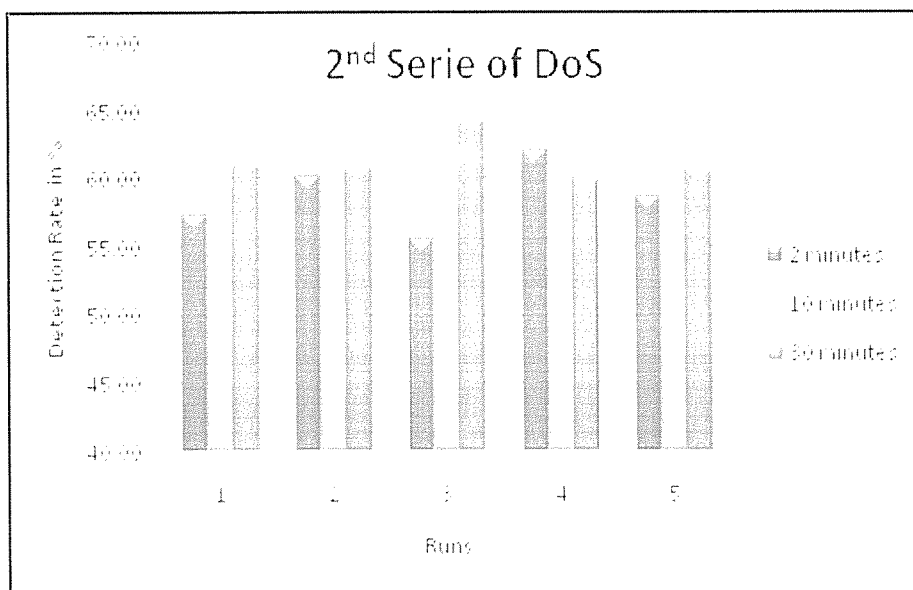


Table 3: 5.1- training window experimental results

Learning period length (minutes)	5	10	30
Overall detection rate	52.1	53.33	55.3
1 st attack detection rate	45.46	45.55	49.39
2 nd detection rate	58.77	61.1	61.2

5.2.2 Findings

Table 5.1 shows that the anomaly-based IDS during the second set of attack has learnt from the previous attack since its detection rate for the second attack is improved by 10 to 15%. The aim of this experiment was to test if the learning window length of The IDS would have an impact on its detection rate. The detection rate has improved by only 1% between 5 minutes and a 10 minutes learning window. The improvement is only 0.3% between 10 minutes and 30 minutes learning windows. In light of these results, it is possible to affirm that SPADE does not need a specific training period length to achieve better detection rates. The last finding from this experiment is the fact that the anomaly detector does not produce any false-positive alarms and detects only two thirds of the packets of a same attack. Section 5.4 provides hypothesis concerning this phenomenon.

5.3 Scenario

After some informal tests following the same exploits steps as the scenario, the behavior of the anomaly detector seemed odd compared to the expected results. Because of this, the scenario experiment was carried out in a more informal manner than the experiment from section 5.2. This odd behaviors was characterized by the fact that SPADE did not produce any false-positive alarms during all the tests and had an unsatisfactory detection rate

5.3.1 Results

Table 5.2 presents an overview of the detection capabilities of each system observed during five runs of each distinct exploit

Exploit	Snort	SPADE
IP scan	Not detected	Not detected
Portscan	Detected	Detected
Brute force	Detected	Not detected
Data theft	Not detected	Not detected

5.3.2 Findings

With regards to the signature-based IDS, Snort, the detector flagged all the exploits for which it had a signature. It detected the portscan and the brute force attack on the FTP server. It did not detect the two other exploits, IP scan and data theft, for several reasons. The reason why it did not detect the IP scan is that, in order to achieve this, Snort needs all the MAC addresses of the subnet in order to detect an Address Resolution Protocol (ARP) scan on all the machines it knows. This difficulty is due to the fact that ARP packets are layer 2 (from the OSI model) packets. Snort works mainly on the Network (3) and the Transport (4) layers, with other methods detecting anomalies in the Data Link Layer and the Application Layer Protocols. The data theft exploit cannot be detected by Snort because it consists of a legitimate FTP file transfer. With regards to the anomaly-based IDS, SPADE, the detector only aged one exploit out of the four in total: the portscan. It detected this exploit with its closed destination preprocessor. These results for the anomaly detector are odd. Given the fact that no FTP activity has been done in the background traffic between the exploit generator and the FTP server prior to carrying out the data theft attack (all the exploits), the anomaly detector should in theory detect the brute force traffic and the data theft traffic as anomalous since the percentage of FTP traffic is increasing when these events occur.

5.4 Recommendation

5.4.1 Experiment

With regards to the experiments conducted in Chapter 5, future work can be organised in two categories. If the aim is to keep the testbed as they are defined in Chapter 4, then three improvements can be brought in: run longer background traffic, run background traffic at lower speed, run SPADE automatic thresholds configuration. This first improvement would be running the entire packet captures from the DARPA set in the first training week or more (there are seven training weeks' data available). This would extend the background traffic cycle considerably. Currently, only the Monday traffic of the first week is used. The second improvement would not prove useful on its own, but extending the length of the background traffic cycle and slowing the speed down could change the results of the experiments. Finally, the last change to the current experimental setup is the most likely to have an important impact on the results. Since SPADE requires to be well tuned to the network it monitors; running its automatic thresholds configurations might adapt the detector to the testbed better than with an "off the box" configuration. If the aim is to change part of the experimental setup, then replacing the captured traffic generation by a realistic traffic simulator is crucial. The other system that would be worth replacing as well is the anomaly detector, SPADE. The only issue with replacing these devices is that the experiment has been implemented with the only currently available tools.

5.5 Area of further research

With regards to the general area of research (IDSs and evaluation), the future work to be considered mainly concerns the evaluation side of IDSs. As shown in Section 2.8, a new way of generating background traffic is needed. Since the DARPA set is now obsolete, a new project involving releasing free recent real network packet captures would bring forward IDS evaluation. The second future work needed is a way of simplifying testbeds creation for IDS evaluation. Currently, any researcher trying to evaluate an IDS often has to setup a complex testbed, composed of diverse tools in order to achieve the evaluation. The problem is that documentation on some of these necessary systems is rare. Often, this

setup process is at least as time consuming as the evaluation process itself. The need for a centralized IDS framework is much needed. Such a framework would provide researchers with all the tools they need to create exploits and different kind of background traffic generation in a centralised device. This would solve the issue of setting up testbeds with many independent tools and would centralise all configuration into one device

5.6 Conclusions

With regards to the training window experiment, the results showed that a learning period variation does not heavily influence the detection rate of SPADE. These results seem odd however. The fact that the anomaly detector SPADE did not generate any false-positives alarms and only detected two thirds of similar intrusive packets is different from any expected results. The results of the scenario experiment proved inconclusive as well. With regards to the signature-based IDS Snort, the results confirmed the results expected in Section 3.5: a signature detector is detecting the intrusions it is set up for Snort configuration had rules about portscans and FTP brute force attacks; hence it detected both steps of a data theft only. The fact that the experiment failed comes from the point-of-view of the anomaly detector. SPADE only detected the second reconnaissance step of the data theft. Given the fact that any traffic between the exploit generator and the FTP server is new to the anomaly detector, it should be flagged as anomalous. To conclude this chapter, it is possible to say that the results of both experiments are inconclusive since the anomaly-based IDS SPADE should be fine tuned to the network monitored. Although the results are inconclusive, the training window experiment showed that SPADE does not need a pure dedicated training period like other anomaly detectors do. The results of the scenario experiment for the signature-based IDS Snort confirmed the fact that signature detectors are as good as their signatures. With regards to the background traffic, a more realistic approach needs to be found in order to run the anomaly detector with traffic over a longer period of time

REFERENCES:

B. Woloch,(2006), “*New dynamic threats require new thinking "moving beyond compliance",*” Computer law & security report, vol. 22, pp. 150 – 156..

C. Gates and C. Taylor, (2007), “*Challenging the anomaly detection paradigm: a provocative discussion,*” in 2006 workshop on New security paradigms. New York, NY, USA: ACM, pp. 21–29

C. Miller,(2007), “*The legitimate vulnerability market,*” in 6th Workshop on the Economics of Information Security, The Heinz School and CyLab at Carnegie Mellon University Pittsburgh, PA, USA, June 7- 8.

D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels,B. Fisher, and B. Fisher,(2007), “*Towards understanding it security professionals and their tools,*” in SOUPS '07 : Proceedings of the 3rd symposium on Usable privacy and security. New York, NY, USA: ACM, , pp. 100–111.

D. K. Smetters and R. E. Grinter,(2002), “*Moving from the design of usable security technologies to the design of useful secure applications,*” in Proceedings of the 2002 workshop on New security paradigms. New York, NY, USA: ACM, pp. 82–89.

M. J. Edwards,(1998), “*Internet Security with Windows NT*”. 29th Street Press,.

J. Graves, W. J. Buchanan, L. Saliou, and J. L. Old, (2006), “*Performance analysis of network based forensic systems for in-line and out-of-line detection and logging,*” in 5th European Conference on Information Warfare and Security (ECIW),.

J. Grossklags, N. Christin, and J. Chuang,(2008), "*Security and insurance management in networks with heterogeneous agents*," in 9th ACM conference on Electronic commerce. New York, NY, USA: ACM, pp. 160–169.

J. Hale and P. Brusil,(2007), "*Security management: A continuing uphill climb*," J.Netw.Syst. Manage., vol. 15, no. 4, pp. 525–553.

K. Buzzard, (1999), "*Computer security - what should you spend your money on?*" Computers & Security, vol. 18, pp. 322 – 334,.

K. Lan, A. Hussain, and D. Dutta,(2003), "*Effect of malicious traffic on the network*," in Passive and Active Measurement Workshop.

K. Ingham and S. Forrest, "A history and survey of network firewalls," University of New Mexico, Tech. Rep., 2002.

K. Labib, "Computer security and intrusion detection," Crossroads, vol. 11, no. 1, pp. 2–2, 2004

H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion detection systems," Computer Network, vol. 31, no. 9, pp. 805–822, 1999.

M. T. Siponen and H. Oinas-Kukkonen, "*A review of information security issues and respective research contributions*," SIGMIS Database, vol. 38, no. 1, pp. 60–80, 2007.

N. I. of Standards & Technology, An Introduction to Computer Security: The NIST Handbook, NIST, Ed. U.S. Department of Commerce, 2006.

P. Fung, L. for Kwok, and D. Longley, "*Electronic information security documentation*," in ACSW Frontiers '03: Proceedings of the Australasian information security workshop

conference on ACSW frontiers 2003. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, pp. 25–31.

F. Gong, “*Deciphering detection techniques: Part ii anomaly-based intrusion detection.*” Network Associates, March 2003.

R. Bace and P. Mell, “*Nist special publication on intrusion detection systems.*” National Institute of Standards and Technology, Tech. Rep., 2001.

R. C. Newman, “*Cybercrime, identity theft, and fraud: practicing safe internet network security threats and vulnerabilities.*” in InfoSecCD '06: Proceedings of the 3rd annual conference on Information security curriculum development. New York, NY, USA: ACM, 2006, pp. 68–78.

S. Northcutt. “*Network Intrusion Detection:*” An Analyst’s Handbook. New Riders. Indianapolis. 1999.

S. Peisert, M. Bishop, and K. Marzullo, “*Computer forensics in forensics.*” SIGOPS Operating Systems Review, vol. 42, no. 3, pp. 112–122, 2008.

S. Perry, “*Network forensics and the inside job.*” Network Security, vol. 2006, pp. 11–13, 2006.

W. H. Baker and L. Wallace, “*Is information security under control ? : Investigating quality in information security management.*” IEEE Security and Privacy, vol. 5, no. 1, pp. 36–44, 2007.

Z.-Q. Wang, H.-Q. Wang, Q. Zhao, and R.-J. Zhang, “*Research on distributed intrusion detection system.*” in 2006 International Conference on Machine Learning and Cybernetics, 2006.

APPENDENCES

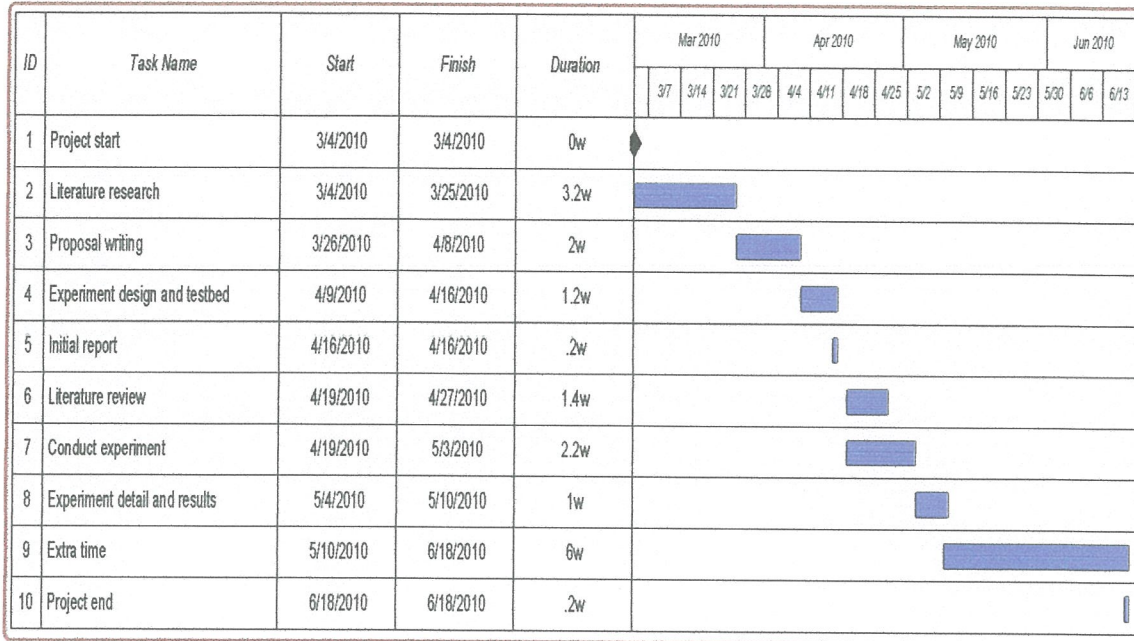
APPENDIX A: INTERVIEW GUIDE

Drafted Interview Guide for the system administrators

**TOPIC: ANALYSIS AND EVALUTION OF NETWORK INTRUSION
DETECTION METHODS FOR IDENTITY THEFT**

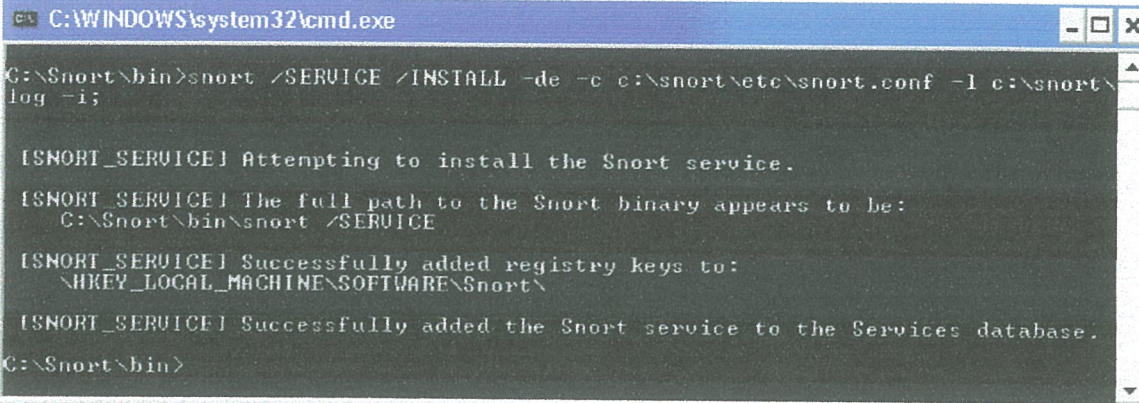
1. What are the most common types of internal threats you face on your network?
2. Do you have a security policy?
3. Do you have a security team?
4. Is log analysis done and if so how often is it done and who does it?
5. Are there instances where staff has escalated their rights such that they have access to documents they should not be accessing?
6. What security configurations have been put in place to protect the internal network?
7. When an attack is detected on given machines, are the compromised machines isolated?
8. What procedures are taken in the event of attack?
9. How often are passwords changed?
10. On average how long will it take an administrator to fix a problem?

APPENDIX B: Gant chart



APPENDIX C: Snort installation configuration and analysis

a) Installation of snort to run as window service

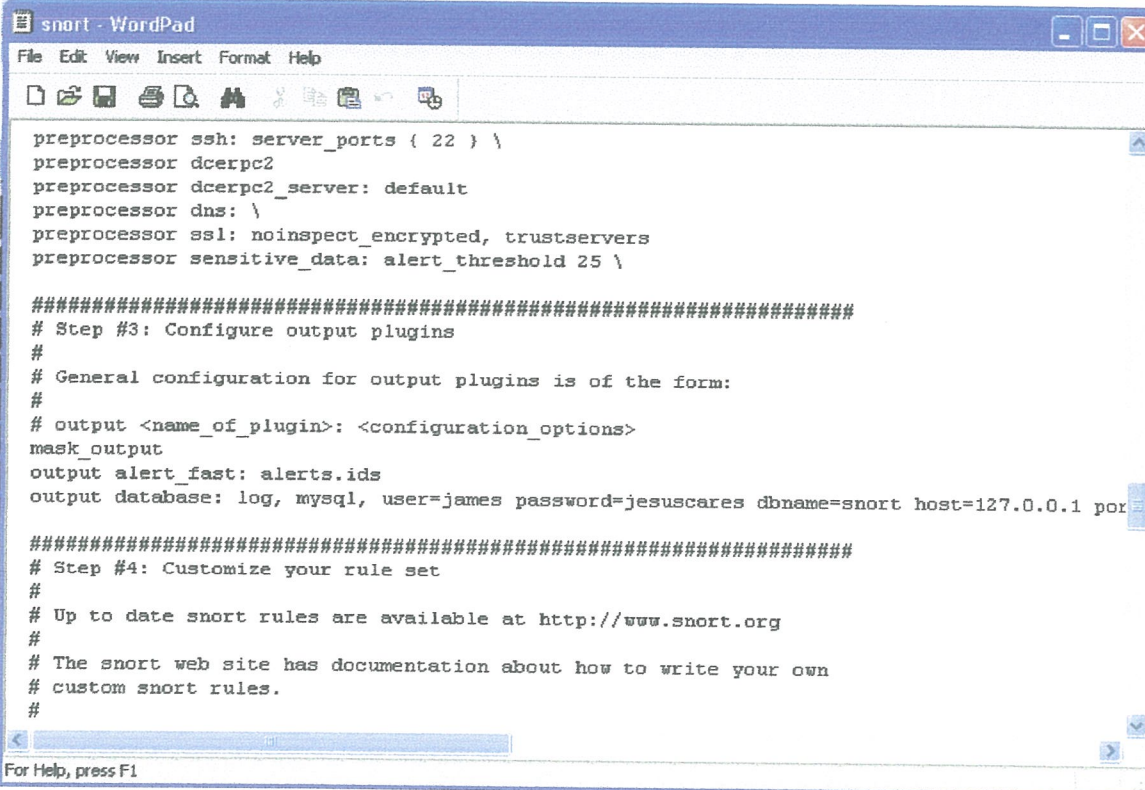


```
C:\WINDOWS\system32\cmd.exe

C:\Snort\bin>snort /SERVICE /INSTALL -de -c c:\snort\etc\snort.conf -l c:\snort\log -i;

ISNORT_SERVICE! Attempting to install the Snort service.
ISNORT_SERVICE! The full path to the Snort binary appears to be:
C:\Snort\bin\snort /SERVICE
ISNORT_SERVICE! Successfully added registry keys to:
\HKEY_LOCAL_MACHINE\SOFTWARE\Snort\
ISNORT_SERVICE! Successfully added the Snort service to the Services database.
C:\Snort\bin>
```

b) Configuration of snort (snort.conf)



```
snort - WordPad
File Edit View Insert Format Help

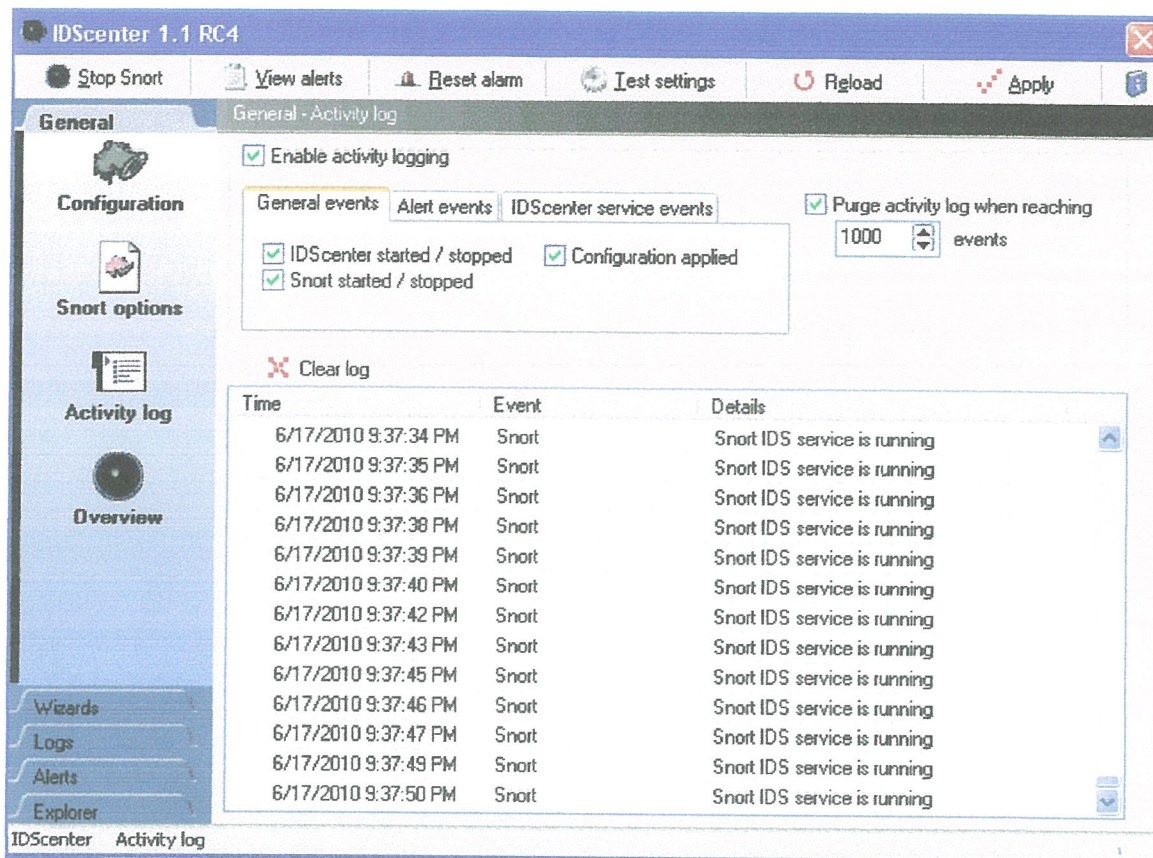
preprocessor ssh: server_ports { 22 } \
preprocessor dcerpc2
preprocessor dcerpc2_server: default
preprocessor dns: \
preprocessor ssl: noinspect_encrypted, trustservers
preprocessor sensitive_data: alert_threshold 25 \

#####
# Step #3: Configure output plugins
#
# General configuration for output plugins is of the form:
#
# output <name_of_plugin>: <configuration_options>
mask_output
output alert_fast: alerts.ids
output database: log, mysql, user=james password=jesuscares dbname=snort host=127.0.0.1 port=3306

#####
# Step #4: Customize your rule set
#
# Up to date snort rules are available at http://www.snort.org
#
# The snort web site has documentation about how to write your own
# custom snort rules.
#

For Help, press F1
```

c) Running and analyzing snort with IDScenter



d) Running snort in Sniffer mode

