Development of Countermeasures against some Internet Security Threats

by

Esan Ayodele Benjamin .C
(10BD000391)

Submitted in Partial Fulfillment of the Conditions
for
the Award of a B.Eng. In Electrical and Information Engineering
to

The College of Science and Engineering

Department of Electrical and Information Engineering

Landmark University, Omu-Aran, Kwara State, Nigeria.

Supervisor: Professor B. J. Olufeagba

Co-Supervisor: A. O. Alimi

June 2015

DEDICATION

This project is dedicated to the Glory of God Almighty for His provision, wisdom, infinite mercies, protection, counsel and grace.

To my dearest mother, my inspiration and first role model, your love, prayers, support and encouragements towards me compares to no other and am so truly grateful for that.

To my siblings for their immense support and care.

To the Department of Electrical and Information Engineering, for the love and team spirit imbibed in me.

And to all those who brought light and direction in my academic journey in Landmark University.

ACKNOWLEDGEMENTS

I acknowledge God's mercy and grace over my life throughout the course of this work, to Him alone be glory, honor and adoration. I also acknowledge the following people who contributed immensely towards the success of this project.

My supervisors, Professor B. J. Olufeagba and Mr. Alimi Adeleke, for their invaluable contributions, encouragements and support towards the attainment of this brilliant end.

My appreciation also goes to my colleagues and friends whose valuable contributions are inestimable, Adejorin Martins, Adenle Praise, Uloko Emmanuel, Olomo Oyindamola, Izundu Chukwudi to mention a few. God bless you all.

I am also grateful to all members and staffs of the department of Electrical and Information Engineering, Landmark University for their various contributions throughout my period of study. Also to Engr. Dickson Egbune who has been like a father to me, am grateful for his willingness to hear me out and help me whenever i needed help.

My sincere appreciation also goes to the chancellor, Dr. David Oyedepo for the privilege given to me to be beneficiary of the scholarship to study in this great institution. You are indeed a savior to this generation and may God Almighty continue to increase His grace and anointing upon you for greater impact and exploits in ministry in Jesus' name.

My heartfelt and deep appreciation goes to my mother, Mrs Isioma Loretta Esan, being a single parent you worked all day, stayed up late, stood in prayers for my destiny. Words can't express how indebted i am to you. I pray you live long enough to reap the benefits of the seeds you have sown in my life and that of my siblings. I love you and God bless you.

ABSTRACT

The internet is an unavoidable infrastructure that can provide enormous benefits to the average person/organization. Unfortunately there are elements whose goal is to interfere with the smooth benefits of it to users for either mischief or criminal intent. This has spawned a bourgeoning industry that focuses on developing and deploying countermeasures to the products of hackers and other internet pirates.

Internet security is configured around several structural features that may be maintained by the user. This work will be focused on identifying all types of entry points into systems, determining the mode of attack, study tactics that can be used to foil such intrusions and fully identify requirements and protocols for effective use of countermeasures and the infrastructure needed for identifying attempted intrusion before it is fully developed.

TABLE OF CONTENT

List of Figures

List of Tables

CHAPTER ONE

INTRODUCTION

1.1 Background of Study

In daily businesses transactions, the use of computer networks is fast approaching a peak level where virtually every human activity requires connection to the web. Before the internet age, business transactions and social relations were often limited by distance, but more recently the internet has provided a means of bridging the communication gap between individuals thus allowing them to share ideas and information more seamlessly.

With the numerous advantages of the internet comes its challenges, the internet has opened up loopholes for persons with dubious intents to compromise networks, enabling them to steal confidential information. Various research and study are being carried out to curtail the activities of these fraudulent persons through different securities tools, however many loopholes exist today that attackers harness to compromise networks, some of which are vulnerabilities in software applications running on servers, poorly configured firewall, unfiltered emails, and complacency on the part of network administrators to the security of their networks.

In 2002, more than 10 million people were victims of identity theft, costing the average victim more than $1,000 and a year's time to repair their credit. More than 95% of Internet users have inadequate protection from online threats. Over 90% of computer users have dangerous "spyware" lurking on their computers without their knowledge. Also in 2002, nearly 20 million people had the skills to hack a computer. In 2003, Internet-related identity theft more than tripled. Today, a typical online PC is "scanned" by outside intruders twelve times every day [1].

Among the top 5 most brutal attacks in 2014 as reported by Forbes, in May, eBay recorded its biggest hack so far, with the personal information of over 233 million users compromised.

Evernote was also attacked, although it was fixed within a couple of hours, but the DoS attack made the application unavailable to legitimate users. Hacking group Rex Mundi held Domino's Pizza to ransom with over 600,000 Belgian and French customer records. In exchange for the personal data, which included names, addresses, emails, phone numbers and even favorite pizza toppings, Mundi demanded $40,000 from the fast-food chain [2].

The stated statistics all point to the fact that cyber-crimes are becoming highly noticeable with potentially undesirable consequences that could neutralize the advantages that make e-transactions so attractive. The only answer to eliminating this is through an organic security approach that can continuously deny access to unauthorized entities.

Network security is so vital that network administrators and users must constantly update and monitor security software. Keeping abreast of news on internet security is also vital while communications logging and monitoring should be mandatory in order to curtail illegal transfers to/from the system.

Linux is a good and proven operating system adapted for networking activities and provides numerous configurations and utilities tailored for optimal network security. It is an open source OS, and very flexible unlike windows, which is both an enterprise/commercial OS and not very flexible. Hence network admins running Linux OS on their server hardware can have full control over the OS, and can make limitless configurations even at the kernel level.

1.2 Problem Statement

The problem addressed in this thesis is to create a setup for exploring weaknesses in the security of simple computer networks and test processes that can be used to frustrate attacks aimed at compromising the network.

1.3 Aim

The aim of the work is to create a model network and a set of countermeasures capable for identifying and frustrating access by unauthorized entities.

1.4 Objectives

The objectives of this research are to:

i. set up a Local Area Network (LAN) in a real/physical environment and then in a virtual environment;
ii. identify and study methods of intrusion that attackers use in compromising networks and;
iii. develop innovative countermeasures to further aid in securing network servers.

1.5 Scope

The scope of this work involves the use of Kali Linux to carry out preventive measures against certain attacks common to recent systems/servers observing their performance over different form of attacks. A few tools adopted by attackers to compromise systems will be considered.

1.6 Methodology

The methodology used in realizing the desired objectives includes:

a. establishing a real/physical LAN comprising a 3-computer network configured with a hub with the third as the intruder running a Kali Linux OS and a virtual counterpart comprising a single host OS and 3-guest windows - Linux OS running in an oracle virtual box to study the impact of possible intrusion;
b. testing each system and studying the effects of intrusion by a number of methods and;
c. implementing counter-measures on the LAN created.

1.7 Thesis Layout

The report comprises of five chapters. Chapter one presents the introduction which gives an overview of the research. Chapter two reviews existing literatures and standards related to the project. Chapter three shows the methodology for setting up network environments that are used to carry out attacks and test countermeasures developed. Chapter four reports the results of attacks and countermeasures obtained from different networks and shows detailed graphical illustrations while chapter five conclude the project work and offers recommendation for future study.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The internet comprises millions of computers from around the world, linked to each other by a network of telephone lines, satellite connections and cables. Securing workstations/servers often requires the harmonious activities of a variety of measures in ensuring that the network on which they reside becomes impenetrable to most if not all intrusion attempts an attacker may use in trying to compromise the network. There are numerous technologies that network administrators utilize in maintaining their networks, the most prominent being the implementation of a firewall. This chapter discusses in details intrusion methods as well as countermeasures against such intrusions with emphasis on adopting a six tier security model comprising of a Two factor authentication, Firewall, PSAD, IDS, SSL enabled webpage, and a local-to-remote backup.

2.2 Concept of Internet Security

The Internet was something only “techies” talked about. It was a new limitless source of information, with very few users. Today, the Internet has already become an essential part of our lives. It’s where we access our banking records, credit card statements, tax returns and other highly sensitive personal information. By the end of this decade, over 2 billion people will be connected to the Internet—that’s about half the world’s current population [1]. But with all the good things the Internet offers us, it also opens the door to serious, potentially devastating threats. Unlike corporate and government computer systems, few personal computers have any safeguards beyond basic virus protection. That means anytime one is online, he/she becomes a potential target for online criminals and hackers. And if you have high-speed Internet access, your computer is online most of the time, making Internet criminals and hackers a 24-hour-a-

day, year-round threat to you, your personal information, and your family. A few of the statistics is given below.

In 2002, more than 10 million people were victims of identity theft, costing the average victim more than $1,000 and a year's time to repair their credit. More than 95% of Internet users have inadequate protection from online threats. Over 90% of computer users have dangerous "spyware" lurking on their computers without their knowledge. In 2002, nearly 20 million people had the skills to hack a computer. In 2003, Internet-related identity theft more than tripled. Today, a typical online PC is "scanned" by outside intruders twelve times every day [1]. Among the goals of internet security is:

- Confidentiality – Prevents unauthorized use or disclosure of information.
- Integrity – safeguards the accuracy and completeness of information.
- Availability – authorized users have reliable and timely access to information.

When referring to network security, a network security triangle analogous to the 'Crime Triangle' model used by law enforcement agents can be adopted to describe the requirement for a network intrusion to occur. The same three criteria must exist before a network security breach can take place. The three "legs" or points of the triangle are shown in the figure below [3]

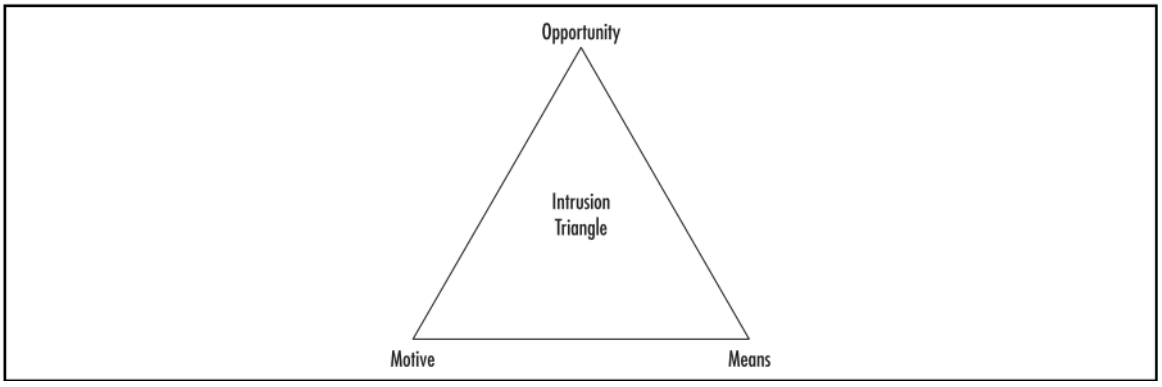

Fig 2.1 Intrusion Triangle

Looking at each of the legs individually, we can see that

- Motive: An intruder must have a reason to want to breach the security of your network (even if the reason is "just for fun"); otherwise, he/she won't bother [3].
- Means: An intruder must have the ability (either the programming knowledge, or, in the case of "script kiddies," the intrusion software written by others), or he/she won't be able to breach your security [3].
- Opportunity: An intruder must have the chance to enter the network, either because of flaws in your security plan, holes in a software program that open an avenue of access, or physical proximity to network components; if there is no opportunity to intrude, the would-be hacker will go elsewhere [3].

Thinking about the three-point intrusion criteria for a moment, it's evident that there is really only one leg of the triangle over which the network administrator or security specialist has any control. It is unlikely that you can do much to remove the intruder's motive or the means he/she uses, the only one thing that the network admin can do is affect the opportunity afforded the hacker [3]. One of the most important, and at the same time most overlooked aspects of a comprehensive network security plan is physical access control. This matter is often left up to facilities managers or plant security departments, or it is outsourced to security guard companies. Network administrators frequently concern themselves with sophisticated software and hardware solutions that prevent intruders from accessing internal computers remotely, while doing nothing to protect the servers, routers, cable, and other physical components of the network from direct access. It is important for you to make physical access control the "outer perimeter" of your security plan and this means [3]:

- Controlling physical access to the servers

- Controlling physical access to networked workstations
- Controlling physical access to network devices
- Controlling physical access to the cable
- Being aware of security considerations with wireless media
- Being aware of security considerations related to portable computers
- Recognizing the security risk of allowing data to be printed out
- Recognizing the security risks involving floppy disks, CDs, tapes, and other removable media

Two categories of controls adopted for network security includes the use of an Intrusion detection system and Honey pots. The IDS is divided into 2 categories; the Host-based intrusion detection systems (HIDS) which uses software's running in the system to monitor the activity of the system itself and to detect signs of malicious activity, a common example being the 'anti-virus software' and the Network-based IDS (NIDS) which uses software's that examines network activities for signs of an intruder. An example of the NIDS is the Signature based network intrusion detection system. However a major weakness of the signature based network intrusion detection system is that the software is unable to detect new attacks if the signature does not match exactly, thus a nifty hacker may modify the attack in some fashion to beat the system. Again another weakness in Honeypots is that although they can be a useful security system, but they are more time and resources consuming as the cost of maintaining the system is not commensurate with the level of security which it provides [4].

2.3 Hackers' Tool Set

There are a couple of tools that are vitally important in any hackers' world. Asides the Operating system of choice used, of which Linux is most preferred because of its flexibility and its open

source nature. Many of these tools are open source and others are commercially available. Some of these tools are as listed below:

2.3.1 Nmap

Nmap is an abbreviation of 'Network Mapper', as is a very well-known free open source hacker's tool. Nmap is used for network discovery and security auditing. Many system admins use nmap for network inventory, open ports, managing service upgrade schedules, and monitoring host or service uptime. The tool uses raw IP packets in creative ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions and possible patches) and what type and version of packet filters/ firewalls are being used by the target [5].

2.3.2 Metasploit

The Metasploit Project is a hugely popular pen-testing or hacking tool that is used by cyber-security professionals and ethical hackers. Metasploit is essentially a computer security project that supplies information about known security vulnerabilities and helps to formulate penetration testing and IDS testing [5].

2.3.3 Angry IP scanner

Angry IP Scanner, also known as 'ipscan' is a freely available (open-source and cross-platform) hacking network scanner that is both fast and easy to use. The main purpose of this hacking tool is to scan IP addresses and ports to find open doors and ports [5].

2.3.4 Ettercap

Ettercap has a huge following and is widely used by cyber-security professionals. Ettercap works by placing the users' network interface into promiscuous mode and by ARP poisoning the target machines, i.e. facilitating a 'Man-In-The-Middle' or MITM attack. Once successful, Ettercap

(and the hacker) can deploy various attacks on the victims. A popular feature about Ettercap is its' ability to support various plugins [5].

2.3.5 John The Ripper

John the Ripper is a popular password cracking pen-testing tool that is most commonly used to perform dictionary attacks. John the Ripper takes text string samples (from a text file, referred to as a wordlist, containing popular and complex words found in a dictionary or real passwords cracked before), encrypting it in the same way as the password being cracked (including both the encryption algorithm and key), and comparing the output to the encrypted string. This tool can also be used to perform a variety of alterations to dictionary attacks [5].

2.3.6 Wireshark

Wireshark is probably the best tool when it comes to sniffing for and collecting data over a network. Wireshark has boosted its capabilities with the support of several types of networks (Ethernet, 802.11, etc.) and also in the simplicity of its use through a very friendly user interface. Beyond the sniffing features, Wireshark is also a great way to validate the network filtering policy. When placed near filtering devices, it can detect the protocols and communication flow in use [6].

2.4 Introduction to Kali Linux

Kali Linux is a debian-derived Linux distribution designed for digital forensics and penetration testing. It is a collection of free penetration testing and security auditing tools, all packaged into a single Linux distribution. Kali Linux contains 300+ tools designed for: Information Gathering, Vulnerability Analysis, Web Applications, Password Attacks, Wireless Attacks, Exploitation, Sniffing/Spoofing, Maintaining Access, Reverse Engineering, Stress Testing, Hardware Hacking, Forensics, and Reporting [7].

While its primary audience is professional penetration testers, it provides the tools for performing password recovery, forensic analysis, and web application testing. Because the tools are preinstalled, maintained and updated, and configured to work together where appropriate, it is an ideal situation for people and organizations that need to do security testing without having the time and resources to maintain their own custom infrastructure. That said, even organizations that have the time and resources may find that they have little need to maintain a custom infrastructure with Kali Linux available. For professional penetration testers, Kali Linux includes password crackers, wireless sniffers, network scanners, and explication tools. For forensic analysis, Kali Linux provides a mode that does not touch the internal hard drive, does not auto mount any removable media, and allows a potentially compromised system to be examined in great detail, along with tools to track the information gathered. For IT departments, Kali Linux has network scanners, vulnerability analysis frameworks, and password recovery tools. With the huge number of tools installed, Kali Linux becomes a very nice tool catalog where users can look through a categorized menu of security tools, making it easy to search for a tool that might meet their needs. And since they are preinstalled, the tools can be quickly evaluated. Many of the tools are text-based and run from the command-line, while others have graphical front ends to make them easier to use. Some provide data intended to be fed into other tools, while others provide detailed reporting and management capabilities [7].

2.5 Intrusion Methods

2.5.1 Information Gathering Attack

Information gathering is the most essential and important task of attackers as knowing the goals and the interests of the target gives valuable information for furthering more severe attacks. These targets could be persons, organizations, network and even web applications. Information

gathering denotes the collection of information before the attack. The idea is to collect as much information as possible about a target which may be valuable later [8]. Among the techniques of Intelligence gathering are; information available in Air [9]. Lists of such types of information are: Archived data of firm, Company website (web pages), Privacy policy used in the application, Security policy used in the application, Client information, Testimonials/Reviews, Exact location detail, Employee information (location, contact, area of Interest, etc.).

Company webpages do sometimes by mistake provide valuable information about their security policy and configuration directly to the client side. Moreover, checking HTML source code for the comments section is a very handy and useful trick because things which are not actually made for the public are easily available via HTML comment tags [9]. There are numerous utilities that could be used by an attacker in cloning a website in order to perform a thorough offline reconnaissance; an example of such utility in Linux is the wget and httrack utility. Checking related organizations is also gives credible information to the attacker. For example, if any IT outsourcing company is there, it's very common to find related organizations about it. Blogs and press releases of such organizations should be checked for any useful information as many other companies and persons will post comments and give reviews [9]. Also using a Google query as shown below, related organizations to the target companies' website can be obtained so the Google query will be like this: *related: www.quora.com* Google will list all other organizations that are related to the *quora.com* company. The information obtained can sometimes be used as a social networking attack, which can be done either directly or indirectly. Physical information can reveal a lot about the target organization. After getting the exact physical location an attacker can then devise various means of attacking target the organization. He/she can perform dumpster diving, that is, checking waste or rubbish papers and posts dropped

from the organization and such material can contain employee names, id numbers, client names etc. All these non-technical methods are also known as no-tech hacking [9]. The best weapon for this information is *Google Earth*. An attacker can also use Google Maps to access the street view and can actually see the streets where the organization is located. Google also collects Wi-Fi information for nearby locations. This collects the information about all Wi-Fi networks in a location along with its MAC address. A Linux tool called *the harvester* can also be used to find out a company's employee information. This tool can provide names of employees working within a company and their respective email ids. Hackers/attackers use this information as usernames in order to gain access to any authorized network, router, etc.

Fig 2.2 Google car collecting Wi-Fi information

Hackers may also use below sources listed below to find phone numbers and the physical addresses of an employee [9]: www.phonenumbers.com, www.411.com, www.yellowpages.com. Information could also be obtained from social networking sites, where people may share their

emotions, enemies, best friends, likes, dislikes, bank details, etc. In addition, people's technical interests and their resumes or career activities can be found on sites like *Linkedin.com, Dice.com, Jigsaw.com, Careerbuilder.com*, etc. [9]. In finding the domain information of a company/organization, along with its administrator's information and registration the WHOIS database provides such and is available at *whois.iana.org* website.

2.5.2 Port Scan Attack

Port scanning is the process of connecting to TCP and UDP ports for the purpose of finding what services and applications are open on the target device. Once open, applications or services can be discovered. At this point, further information is typically gathered to determine how best to target any vulnerabilities and weaknesses in the system [10]. When an attacker wants to gain access into any remote system, he/she often looks for a vulnerability either in the operating system on which the remote system runs, or in the applications running on specific ports on the remote system. Using tools such as nmap, amap, meterpreter, and an attacker can perform various types of port scans on the target system and consequently harvest lots of useful information. A port scan doesn't directly damage the system, but it potentially helps an attacker in locating ports that are available and vulnerable to launch attacks. There are 65535 standardly defined ports on a computer, and they are categorized into 3 large ranges which are [11]:

a) Well Known ports(0-1023)
b) Registered ports(1024-49151) and
c) Dynamic and/or private ports (49152-65535).

Generally, these port scan could be either noisy (aggressive) or quiet (stealth). Some example of noisy scans with nmap includes:

a) *TCP connect () scan (sT):* In this scan, the attacker establishes a full TCP 3-way handshake with the target. In this kind of scan, it is impossible for the attacker to go undetected, as the scan is logged on the target machine.

b) *TCP ping scan (PN):* This pings or sends an icmp packet to the entire targets it probes. Because an icmp response is to be expected, the target machine gets to know that an attacker is requesting to know if it is up and running

c) *OS fingerprinting (O):* nmap can detect the kind of operating system running on the target machine by sending special packets to the target and analyzing the response packets, it checks the fields in the header packet and makes intelligent guesses of the type of operating system running on the target machines.

d) Version scan (sV): nmap uses this option, to detect the version of applications running the ports being scanned.

2.5.3 Sniffing Attack

Sniffers are basically "Data interception" technologies. Most networks use broadcast technology where messages from one node on the network can be read by another node on that network. In reality, all other nodes on the network except the node for which the information is meant for ignores or discards the packet. However, computers can be made to accept messages even if they are not meant for them; this is what is referred to as sniffing. This often requires the computer to be put in promiscuous mode which usually needs root or administrative privileges.

Sniffing involves capturing, decoding, inspecting and interpreting the information inside a network packet on a TCP/IP network. The purpose is to steal information, usually user IDs, passwords, network details, credit card numbers, etc. Sniffing is generally referred to as a “passive” type of attack, wherein the attackers can be silent/invisible on the network. This makes

it difficult to detect, and hence it is a dangerous type of attack [12]. The sniffing process is usually used by hackers either to get information directly or to map the technical details of the network in order to create a further attack. Hackers are always in favor of sniffing, because it can be done for a longer time without getting caught [12]. There are two basic types of Ethernet environment as recorded in [13] and how these network sniffers works in both cases differ. These Ethernet environments are;

a) *Shared Ethernet*: In a shared Ethernet environment, packets meant for one machine are received by all the other machines. If 4 hosts are connected to the same bus, thus when a machine (host 1) wants to talk to another machine (host 2) in such environment, it sends a packet on the network with the destination Mac address of host 2 along with its source MAC address. All the computers on the shared Ethernet (host 3 and 4) compare frame's destination MAC address with their own. If the two don't match, the frame is quietly discarded. A machine running a sniffer breaks this rule and accepts all frames. Such a machine is said to have been put into promiscuous mode and can effectively listen to all the traffic on the network. Sniffing in a Shared Ethernet environment is totally passive and hence extremely difficult to detect [13]. Wireless networks are a form of shared Ethernet and as such sniffing is mostly prevalent in these networks as nodes connect to a particular SSID broadcasted by a wireless radios .

b) *Switched Ethernet*: An Ethernet environment in which the hosts are connected to switch instead of a hub is called a switched Ethernet. The switch maintains a table keeping track of each computer’s MAC addresses and the physical port on the switch to which that MAC address is connected and delivers packets destined for a particular

machine correspondingly. The switch is an intelligent device that sends packets to the destined computer only and does not broadcast it to all the machines on the network, as in the previous case. This results in better utilization of the available bandwidth and improved security. Hence the process followed earlier; of putting the machine into 'promiscuous mode', to gather packets does not work [13].

However a switched network is not completely secure from sniffing, as there are other mean of sniffing on the network besides putting the network interface card in promiscuous mode. These methods are; Arp spoofing & Mac flooding [13].

2.5.4 Denial of Service Attack

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) the service or resource they expected [14]. Victims of DoS attacks often target the web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle [14].

Denial-of-service attacks come in a variety of forms and aims at a variety of services of which there are three basic types of attack [15]:

- consumption of scarce, limited, or non-renewable resources
- destruction or alteration of configuration information
- physical destruction or alteration of network components

2.5.4.1 Consumption of Scarce Resources: Computers and networks need certain things to operate such as network bandwidth, memory and disk space, CPU time, data structures, access to other computers and networks, and certain environmental resources such as power, cool air, or even water. Denial-of-service attacks are most frequently executed against network connectivity of which the goal is to prevent hosts from communicating on the network. An example of this type of attack is the "SYN flood" attack. In this type of attack, the attacker begins the process of establishing a connection to the victim machine, but does it in such a way as to prevent the ultimate completion of the connection. In the meantime, the victim machine has reserved one of a limited number of data structures required to complete the impending connection. The result is that legitimate connections are denied while the victim machine is waiting to complete bogus "half-open" connections. In this case, the intruder is consuming kernel data structures involved in establishing a network connection [15]. An intruder may also be able to consume all the available bandwidth (bandwidth consumption) on your network by generating a large number of packets directed to your network. Typically, these packets are *ICMP ECHO* packets, but in principle they may be anything. Further, the intruder need not be operating from a single machine; he may be able to coordinate or co-opt several machines on different networks to achieve the same effect [15].

2.5.4.2 Destruction or Alteration of Configuration Information: An improperly configured computer may not perform well or may not operate at all. An intruder may be able to alter or destroy configuration information that prevents you from using your computer or network. For example, if an intruder can change the routing information in your routers, your network may be disabled. If an intruder is able to modify the registry on a Windows NT machine, certain functions may be unavailable [15].

2.5.5 Arp-spoofing Attacks

ARP spoofing is the technique of forging fake ARP messages on a network. It is possible to update a host's ARP cache with false information via spoofed ARP Replies. This technique is known as 'ARP Poisoning' and is the basis of more complex attacks [16]. A node on an IP/Ethernet network maintains two addresses. The first is the address of the network interface card (NIC) which is the hardware address known as the Media Access Controller (MAC) address. The MAC address in theory is globally unique, as it is comprised of the *manufacturer's serial number* and *model number*. It is also supposed to be static, as it is stored in the firmware of the card itself. This MAC address is required to send frames of data out on an ethernet network. When a node sends a frame to another node the source and destination MAC addresses are held in the ethernet header of the frame. The second address a node on an IP/Ethernet network has is its IP address. An IP address is a unique virtual address that is assigned via software and bound to a hardware address, e.g. a NIC. IP communicates by constructing packets of data, encapsulating any higher protocols (e.g. TCP or UDP) and transmitting them via the network layer, in our case ethernet frames. When the ethernet frame is built to send an IP packet the ethernet header must be filled in, however ethernet will not know the destination MAC address that is bound to the destination IP address. A method of resolving this address must be available to ethernet and it comes in the form of the Address Resolution Protocol. *ARP sole purpose is to resolve 32-bit logical IP addresses into their associated 48-bit Ethernet hardware address* [16]. ARP communicates via four messages. An ARP Request is a message asking to resolve a given IP address into its bound MAC address. This message is usually broadcast to all hosts on a network via the ethernet broadcast address. Every receiving host will examine the request to see if it is assigned the specified IP address and if so will respond with an ARP Reply telling the

requesting host its MAC address. Two more messages exist, a Reverse ARP (RARP) Request and a RARP Reply. A RARP Request asks to resolve a given MAC address into its associated IP address. A RARP Reply is the response to a RARP Request giving the IP address of the associated MAC address. Often hosts maintain a cache of ARP replies to minimize the amount of ARP requests being broadcast. When a host receives an ARP reply it will update this cache with the new IP address to MAC address association. This will happen regardless of whether the host initially sent out an ARP Request, due to ARP being a stateless protocol [16].

2.6 Countermeasures

2.6.1 Two factor authentication (2FA)

Two-factor authentication is a security process in which the user provides two means of identification, one of which is typically a physical token, such as a card, and the other of which is typically something memorized, such as a security code [17]. An attacker may occasionally break an authentication factor in the physical world. A persistent search of the target premises, for example, might yield an employee card or an ID and password in an organization's trash or carelessly discarded storage containing password databases. If additional factors are required for authentication, however, the attacker would face at least one more obstacle. The majority of attacks come from remote internet connections. 2FA can make distance attacks much less of a threat because accessing passwords is not sufficient for access and it is unlikely that the attacker would also possess the physical device associated with the user account. Each additional authentication factor makes a system more secure. Because the factors are independent, compromise of one should not lead to the fall of others [17].

To provide an everyday example: an automated teller machine (ATM) typically requires two-factor verification. To prove that users are who they claim to be, the system requires two items:

an ATM smartcard (application of the possession factor) and the personal identification number (PIN) (application of the knowledge factor). In the case of a lost ATM card, the user's accounts are still safe; anyone who finds the card cannot withdraw money as they do not know the PIN. The same is true if the attacker has only knowledge of the PIN and does not have the card. This is what makes two-factor verification more secure since there are two factors required in order to authenticate [18].

2.6.2 Firewalls

A firewall is an integrated collection of security measures designed to prevent unauthorized electronic access to a networked computer system. A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another. To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed to filter incoming or outgoing traffic based on a predefined set of rules called firewall policies [19].

Packets flowing through a firewall can have one of three outcomes:

- Accepted: permitted through the firewall
- Dropped: not allowed through with no indication of failure
- Rejected: not allowed through, accompanied by an attempt to inform the source that the packet was rejected

Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used such as: TCP or UDP, the source and destination IP addresses, the source and destination ports, the application-level payload of the packet (e.g., whether it contains a virus) [19]. There are two fundamental approaches to creating firewall policies (or rule-sets) to effectively minimize vulnerability to the outside world while

maintaining the desired functionality for the machines in the trusted internal network (or individual computer).

1) *Blacklist approach*: All packets are allowed through except those that fit the rules defined specifically in a blacklist. This type of configuration is more flexible in ensuring that service to the internal network is not disrupted by the firewall, but is naïve from a security perspective in that it assumes the network administrator can enumerate all of the properties of malicious traffic [19].
2) *Whitelist approach*: A safer approach to defining a firewall rule-set is the default-deny policy, in which packets are dropped or rejected unless they are specifically allowed by the firewall [19].

Firewalls can be classified into three basic categories: packet filters, proxy servers (which include application gateways and circuit-level gateways), and stateful packet filters. There is a fourth category that is essentially a hybrid of the three main categories. Figure below illustrates the different types of firewalls [20].

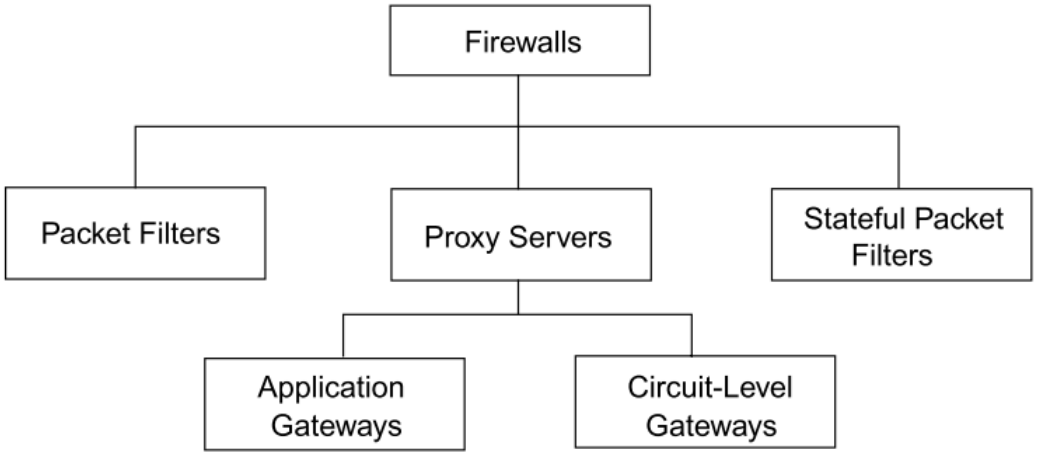

Fig 2.3 Classification of Firewalls

2.6.2.1 Packet Filters

A packet filter is a firewall that inspects each packet for user-defined filtering rules to determine whether to pass or block it. For example, the filtering rule might require all Telnet requests to be dropped. Using this information, the firewall will block all packets that have a port number 23 (the default port number for Telnet) in their header. Filtering rules can be based on source IP address, destination IP address, Layer 4 (that is, TCP/UDP) source port, and Layer 4 destination port. Thus, a packet filter makes decisions based on the network layer and the transport layer. Packet filters are fast and can be easily implemented in existing routers. Unfortunately, they are the least secure of all firewalls. One disadvantage of packet filters is that they have no logging facility that can be used to detect when a break-in has occurred. Also, a packet filtering firewall grants or denies access to the network according to the source and destination addresses and the source and destination ports. Unfortunately, these ports can be spoofed [20].

2.6.2.2 Proxy Servers

A proxy service is an application that redirects users' requests to the actual services based on an organization's security policy. All communication between a user and the actual server occurs through the proxy server. Thus, a proxy server acts as a communications broker between clients and the actual application servers. Because it acts as a checkpoint where requests are validated against specific applications, a proxy server is usually processing intensive and can become a bottleneck under heavy traffic conditions. Proxy servers can operate at either the application layer or the transport layer. Thus, there are two classes of proxy servers: application gateways, which operate at the application layer; and circuit-level gateways, which operate at the transport layer [20].

2.6.2.3 Application Gateways

An application gateway is a proxy server that provides access control at the application layer. It acts as an application-layer gateway between the protected network and the untrusted network. Because it operates at the application layer, it is able to examine traffic in detail and, therefore, is considered the most secure type of firewall. It can prevent certain applications, such as FTP, from entering the protected network. It can also log all network activities according to applications for both accounting and security audit purposes [20].

2.6.2.4 Circuit-Level Gateways

A circuit-level gateway is a proxy server that validates TCP and UDP sessions before allowing a connection or circuit through the firewall. It is actively involved in the connection establishment and does not allow packets to be forwarded until the necessary access control rules have been satisfied [20].

2.6.2.5 Stateful Packet Filters

Although the application gateway provides the best security among the preceding firewalls, its intensive processing requirement slows down network performance. A stateful packet filtering gateway attempts to provide tight security without compromising performance. Unlike the application gateway, it checks the data that passes through at the network layer but does not process it. The firewall maintains state information for each session, where session states include a combination of communication phase and the endpoint application state. When a stateful packet filtering gateway receives a data packet, it checks the packet against the known state of the session. If the packet deviates from the expected session state, the gateway blocks the rest of the session [20].

2.6.3 Port Scan Attack Detector

PSAD is a collection of three lightweight system daemons (two main daemons and one helper daemon) that run on Linux machines and analyze iptables log messages to detect port scans and other suspicious traffic. A typical deployment is to run psad on the iptables firewall where it has the fastest access to log data [21]. PSAD incorporates many signatures from the Snort intrusion detection system to detect probes for various backdoor programs (e.g. EvilFTP, GirlFriend, SubSeven), DDoS tools (mstream, shaft), and advanced port scans (FIN, NULL, XMAS) which are easily leveraged against a machine via nmap. When combined with fwsnort and the Netfilter string match extension, psad is capable of detecting many attacks described in the Snort rule set that involve application layer data [21]. PSAD is developed around three main principles which are:

- Good network security starts with a properly configured firewall [21].
- A significant amount of intrusion detection data can be gleaned from firewalls logs, especially if the logs provide information on nearly every field of the network and transport headers [21].
- Suspicious traffic should not be detected at the expense of trying to also block such traffic [21].

2.6.4 Intrusion Detection Systems

An Intrusion Detection System (IDS) is a network security technology originally built for detecting vulnerability exploits against a target application or computer. The IDS is also a listen-only device, it monitors traffic and reports its results to an administrator, but cannot automatically take action to prevent a detected exploit from taking over the system. Attackers are capable of exploiting vulnerabilities very quickly once they enter the network, rendering the IDS

an inadequate deployment for prevention device [22]. An IDS need only detect threats and as such is placed out-of-band on the network infrastructure, meaning that it is not in the true real-time communication path between the sender and receiver of information [22]. Among the types of IDS available are:

- Network-based IDS
- Host-based IDS

The Host-based IDS uses OS auditing mechanisms: e.g. BSM in Solaris logs all direct and indirect events generated by a user, Monitoring user activities by analyzing shell commands, Monitoring executions of system programs, e.g. sendmail's system calls [23].

Advantages of Host-based IDS as described by [23] are:

- Can detect attacks that cannot be seen by NIDS.
- Can operate in an environment in which network traffic is encrypted.
- Unaffected by switched networks.
- Can help detect Trojan horse or other attacks that involve software integrity breaches

Disadvantages of Host-based IDS as described by [23] are:

- Since at least the information sources reside on the host targeted by attacks, the IDS may be attacked and disabled as port of the attack
- Are not well suited by detecting network scans or other such surveillance that targets an entire network
- Since they use the computing resources of the hosts they are monitoring, therefore inflicting a performance cost on the monitored systems.

However, the Network Intrusion Detection Systems (NIDS) Looks at IP header as well as data parts of every packet on the network and monitors such packets for anomalies. Disadvantages of the Network-Based IDS as described in [23] include:

- NIDS may have difficulties processing all packets in a large or busy network and therefore, may fail to recognize an attack launched during periods of high traffic.
- Modern switch-based networks make NIDS more difficult: Switches subdivide networks into many small segments and provide dedicated links between hosts serviced by the same switch. Most switches do not provide universal monitoring ports
- NIDS cannot analyze encrypted information.
- Most NIDS cannot tell whether or not an attack was successful.

2.6.5 Secure Socket Layer (SSL)

SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted links between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook). SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely. Normally, data sent between browsers and web servers is sent in plain text—leaving you vulnerable to eavesdropping. If an attacker is able to intercept all data being sent between a browser and a web server they can see and use that information [24].

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms. Web browsers such as Internet Explorer,

Firefox and Chrome also display a padlock icon in the address bar to visually indicate that a HTTPS connection is in effect [25]. HTTPS pages typically use one of two secure protocols to encrypt communications - SSL (Secure Sockets Layer) or TLS (Transport Layer Security). Both the TLS and SSL protocols use what is known as an 'asymmetric' Public Key Infrastructure (PKI) system. An asymmetric system uses two 'keys' to encrypt communications, a 'public' key and a 'private' key. Anything encrypted with the public key can only be decrypted by the private key and vice-versa. As the names suggest, the 'private' key should be kept strictly protected and should only be accessible the owner of the private key. In the case of a website, the private key remains securely ensconced on the web server. Conversely, the public key is intended to be distributed to anybody and everybody that needs to be able to decrypt information that was encrypted with the private key [25]. When an HTTPS connection to a webpage is requested, the website will initially send its SSL certificate to your browser. This certificate contains the public key needed to begin the secure session. Based on this initial exchange, your browser and the website then initiate the 'SSL handshake'. The SSL handshake involves the generation of shared secrets to establish a uniquely secure connection between yourself and the website. When a trusted SSL Digital Certificate is used during a HTTPS connection, users will see a padlock icon in the browser address bar. When an Extended Validation Certificate is installed on a web site, the address bar will turn green [25].

2.6.6 Backup (Rsync)

A common system task is backing up files – that is, copying files with the ability to go back in time and restore them. The important thing to remember is that backups are all about copies of the data at a certain point in time [26]. In contrast to backing up is “replication.” A *replica* is simply a copy of the data when the replication took place. By definition, replicas can use a great

deal of space, because each time a replica is made, the entire filesystem as it exists is copied. If you keep several replicas, you are going to have multiple copies of the same data. For example, if you have a 100TB filesystem that is 20% full, (20TB) and you create a copy, you copy 20TB of data. If you make another copy later when the filesystem uses about 25TB (5TB of data has changed), you copy 25TB of data. If you make a third copy of the data at 30TB (another 5TB of changed data), you copy another 30TB. Through time, then, you have had to use 75TB of space (20 + 25 + 30), with no doubt lots of duplicated data wasting space and money [26]. The backup world uses a few techniques that differentiate it from replication. The first is called a *"full backup"* and really is a copy of the data as it existed when the copy was made (point in time). The second backup technique is called an "incremental backup," which only stores the data that has changed since some point in time (typically the previous backup). This technique can save a large amount of storage capacity because files that have not changed are not backed up [26].

Rsync is a sweet utility for syncing files/folders. Many times it is used for producing incremental backups since it is capable of detecting what files are added and changed to a folder. It usually does this by timestamps but it can be set to determine file changes with a more precise (but slow) method using md5 hashes. However, generating md5 hashes for detecting file changes is usually not required [27]. Although the description is accurate, rsync is much more than a simple copy tool. It can copy files both locally, to/from a remote host using a remote shell (e.g., SSH), or to/from a remote rsync daemon. One of the things that make rsync unique is that it has a "delta transfer" capability that reduces the amount of data actually transferred. This capability lets it make replicas, copies, and even backups [27].

2.7 Review of Related Countermeasures against Network Intrusions

In [28] the author proposes the use of PortVis, a tool designed for scan detection, uses summarized network traffic for each protocol and port for a user-specified time period. The summaries include the number of unique source addresses, the number of unique destination addresses, and the number of unique source-destination address pairs. However it is unclear how well this algorithm scales to larger network

The Slow port scan detection method as proposed in [29] detects slow port scanning attacks in which the IDS would be required to capture the traffic for a larger period of time. However, it can be shown that this approach incurs a large delay on the server side and therefore the client will witness a huge degradation in the Quality of Service (QoS). Furthermore, an IDS operating on a large time window might become a target for a Denial of Service (DoS) attack by overloading the server with too many traffic that requires a lot of processing.

The authors in [30] Chen and Cheng present a novel and fast port scan detection method based on partheno-genetic algorithms (PGA). The method can efficiently discover ports that are open most often. During genetic evolution, ports with more open times survive to the next generation with higher probabilities. This approach demonstrates that PGA-based port scan is efficient in average as well as worst cases. Sequential port scans are better in best cases only.

The author in [31] suggests the use of ARP detection technique in which an arp packet is sent to every host on the network and ARP packet is configured such that it does not have broadcast address as destination address and if some host respond to such packets, then those host have their NIC set into promiscuous mode. The author also suggests the RTT (Round Trip Time) detection technique in which the time that the packet takes to reach the destination along with the time which is taken by response to reach the source is observed. The idea behind this technique

is that RTT measurement increases when the host is in promiscuous mode, as all packets are captured in comparison to host that is in normal mode. The drawback of the ARP detection technique is that Windows is not an open source OS, so we can't analyze its software filter behavior as we do in Linux.

The author in [12] suggested two ways to detect sniffers on a network, which are host-based and network-based. In host-based detection, small utilities are used to detect if the NIC is running in a promiscuous mode on any host in a network. Since the basic requirement for a sniffer to work is to put the network interface in "read all" mode, disabling it can very effectively help shutting down stray sniffers but the drawback of this is that most networks deployed in large scale organizations use Wi-Fi connections which are often not protected and hence any stranger within the vicinity of the network coverage can connect to your network and sniff packets.

In the case of network-based detection, anti-sniffer software can be run to detect the presence of specific signature packets, the drawback being that often time's signature packets can be modified by hackers and may go undetected by the anti-sniffer software.

In guarding against DoS/DDoS attacks, the author in [32] suggested that service providers can increase redundancy of network and service infrastructure. However redundancy solutions are not always effective against DoS attacks due to the additional computing resources required to handle incoming traffics which could be costly to acquire and maintain.

Ai-zeng Qian [33] proposed a technique to prevent ARP spoofing by using static ARP entries; however the technique doesn't work with dynamic networks using DHCP addressing as the administrator must assign all IP addresses along with their MAC to the server so it will be not visible for large scale network.

The author in [34] suggested using snort IDS and static ARP entries to solve the ARP problem. Yet, it still needs the administrator to add the static mappings manually. Also, it works only in static networks.

The authors in [35] proposed as a replacement for the ARP protocol the use of the Secure ARP Protocol (S-ARP). The S-ARP protocol is a permanent solution to ARP spoofing but its biggest drawback is that changes will have to be made to the network stack of all the hosts and this is not very scalable as going for a stack upgrade across all available operating systems is something both vendors and customers will not be happy about. Also S-ARP uses Digital Signature Algorithm (DSA) and so have the additional overhead of cryptographic calculations. Although the authors of the paper have claimed that this overhead is not significant.

This study as explained doesn't use only one technology to protect a network but uses a six tier network model to protect and guard against different forms of intrusions as described above in this chapter.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

Access into a system involves due diligence on the part of a hacker as the procedure for compromising networks involves information gathering, network scanning, vulnerability scanning, exploitation and protecting privileges. Following these procedures, an intruder can gain access if the network in question is not adequately fortified against intrusions.

The aim of this work is to create a typical network prototype used in an organization, identify possible loop-holes and subsequently design an operational secure network which will be adequately fortified against the common intrusion methods hackers' employ in penetrating such system. To realize this goal, a comprehensive study of a modeled network created in a real and virtual environment is reviewed, common intrusion means of access to network is also studied as discussed in chapter two and a firewall along-side other countermeasures are designed and incorporated into the modeled network to prevent such attacks. The use of proxy website to access a block site from a server is now becoming prevalent in access network. Method of blocking such proxy is also reviewed and presented.

3.2 A Modeled LAN with Internet Access.

A typical network design is shown in figure 3.1. In creating this network, a Linux system was configured as a router, by utilizing the NAT (Network Address Translation) feature of IPTABLES using the following command, the Linux system can route internal traffic to external addresses 'iptables –t nat –p tcp –A POSTROUTING –s 10.0.9.0/24 –o eth0 –j MASQUERADE'. The output interface represents a connection to the internet and was realized using an android device with internet connectivity, then a hotspot was created through which the

wireless network interface of the router was connected – represented as wlan0. The hub is a five port device and interconnects all the internal devices together.

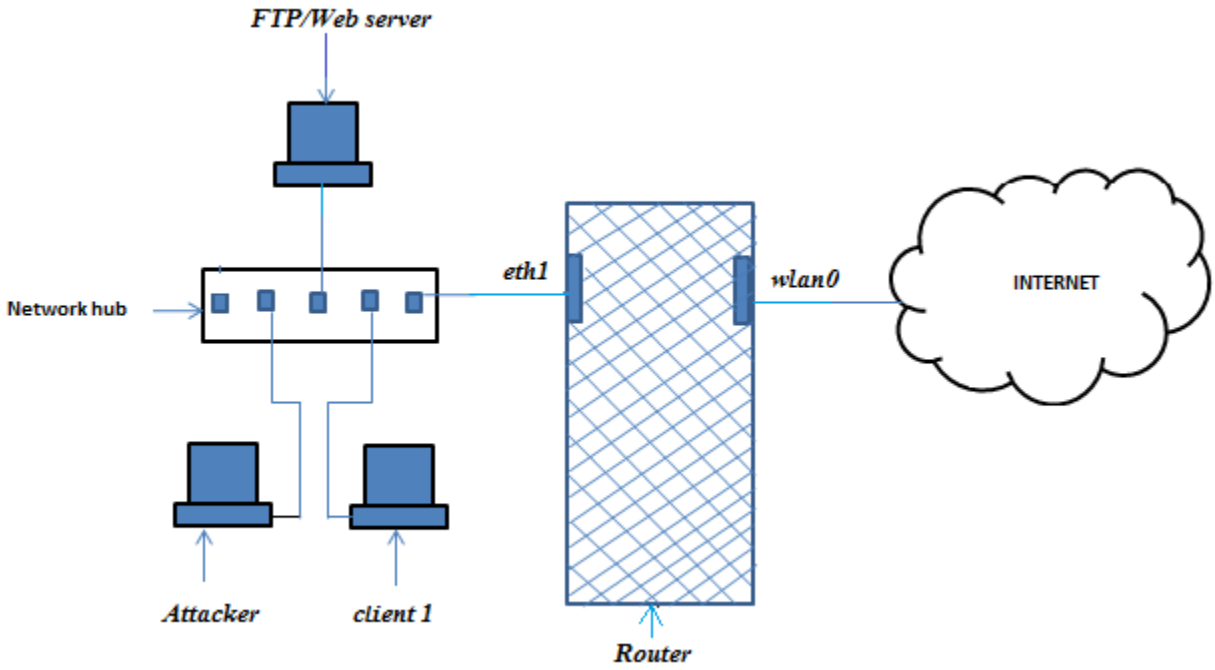

Fig 3.1 Vulnerable Network Design

The network information for fig 3.1 is shown below

Table 3.1 Network Information for modeled network.

Network Nodes	(Type, address)
Router	Eth1 : 10.0.9.2/24 Wlan0 : As assigned by ISP
Client 1	10.0.9.3/24
Attacker	10.0.9.6/24

FTP/Web server	10.0.9.4/24

Using the above network structure, a number of vulnerabilities can be identified among which are the:

- Easy access to information about available and possibly vulnerable services (ports) running on the network from port scans made possible because of the lack of port scan attack detectors.
- Prevalence of sniffing attacks mainly due to the use of a shared network media (Hub) whose inherent weakness is its having a single collision and broadcast domain.
- Access to protected web pages, due to the absence of firewalls and inability to configure web server security features for web pages.
- Prevalence of a Denial of Service (DoS) attacks on the webserver due to the absence of a firewall.
- Prevalence of brute force attacks against the ftp server also due to the absence of a firewall.
- Arp spoofing attacks (MitM) against clients on the network.

Five of the network vulnerabilities as listed above would be tested on the modeled network, with the addition of information gathering attacks (which is actually the first step an attacker takes in penetrating a network) conducted on a university's website to demonstrate how social engineers could harness the information obtained for manipulating individuals to surrender confidential information.

3.3 Attack Strategies

3.3.1 Information Gathering (Reconnaissance)

The procedure for carrying out reconnaissance on the university's website involves getting the detail of the domain registrar for the university. In performing this reconnaissance, an internet connection is needed. A URL exists (https://www.whois.net) in which the domain name of any organization is entered and the resulting domain information is displayed for such organization. It should be noted that reconnaissance could be passive or active. The method used in performing this reconnaissance is passive as its not directly relating to the website.

3.3.2 Sniffing

Network sniffing is peculiar only to local area networks, and usually involves an attacker using a network sniffer to capture information from other clients on the network. The attacker may choose to disguise the mac-address (mac-address spoofing) of his pc so he/she goes undetected by the network administrator. In the network model above, the attacker uses a network packet analyzer – wireshark to capture packets from clients belonging to the same network. The attacker in the model network above captures the ftp username and password of a client logging into the ftp server.

3.3.3 Port Scans

The above network model runs only two services on its servers i.e. a web and a file service and so would be insufficient in ascertaining the severity of this form of attack. Hence a virtual network environment is set up on a Linux pc using the *oracle virtual box* application. The virtual network is as represented in fig 3.2 below. In setting up the virtual environment in Linux, the oracle virtual box was used. It can be downloaded using the following command on the terminal 'apt-get install dkms virtualbox' where dkms is the dependency required for virtualbox to install

without any troubles. After it is downloaded, new guest OS can be added following the graphical menu of the virtual machine.

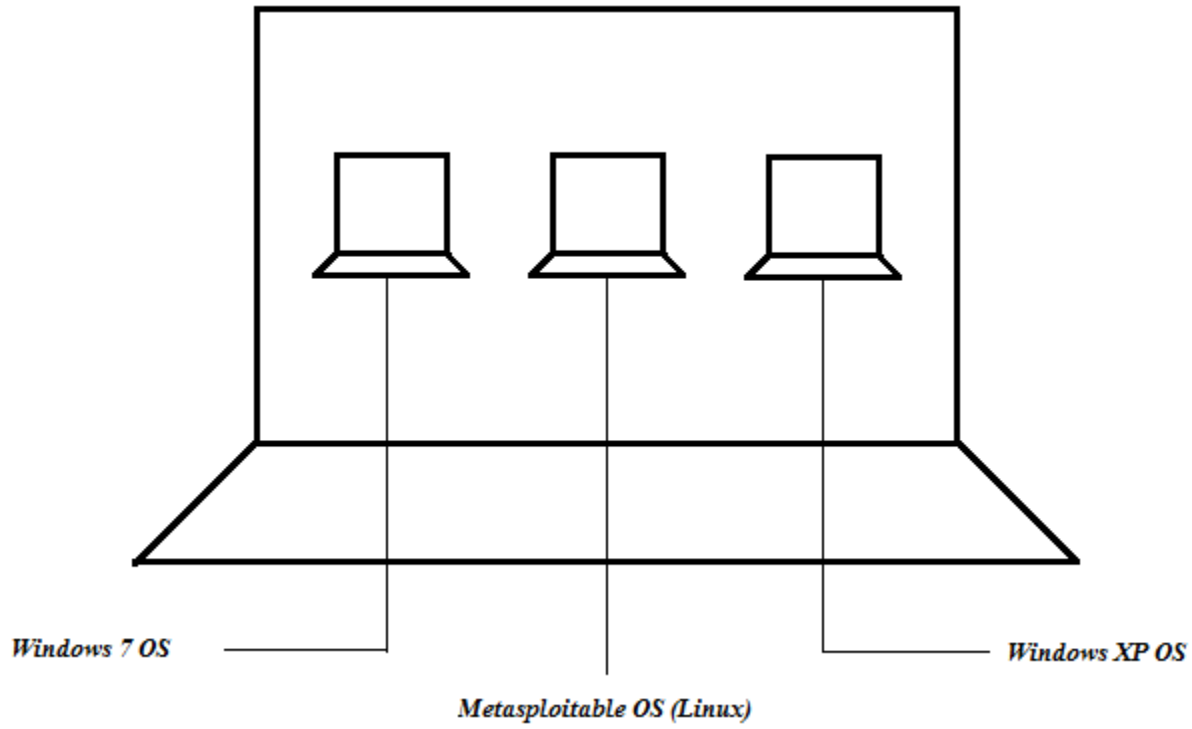

Fig 3.2 Virtual Setup for a 3-Port Scan Attack Testing

The network information for the above network setup is as shown in table below

Table 3.2 Network Information for virtual machine.

Operating System	Network Information
Host OS (Kali Linux)	192.168.50.100/24
Guest OS 1 (Windows XP)	192.168.50.102/24
Guest OS 2 (Windows 7)	192.168.50.103/24
Guest OS 3 (Metasploitable Linux)	192.168.50.101/24

A port scan from the host pc (kali) against each of the guest pc using a port scanner (nmap) is performed. Due to the versatility of the nmap tool, various options abound to aid attackers in port scans.

3.3.4 ARP Spoofing

Arp spoofing is also peculiar to a local area network, also called a man-in-the-middle attack. In intercepting communication between clients on a network, arp-spoofing is used. The attacker (10.0.9.6) uses the arp-spoof tool (arp-spoof or ettercap) available on his Linux pc to perform the MitM attack on the victim (10.0.9.4) and the gateway (10.0.9.2). With this intrusion, the attacker could terminate a connection, intercept and sniff a packet and even alter a packet illegally.

3.3.5 Denial of Service (DoS)

While other intrusions by attackers attempt to steal confidential information from a target, the DoS attack seeks to use up network resources thereby making it inaccessible to other legitimate clients that need to communicate with the network. The attacking machine uses the hping3 tool available in Linux to perform an internal DoS attack against the webserver in the modeled network above.

From the intrusion tests as described above, it's seen that the modeled network design is flawed, hence making numerous vulnerabilities available to attackers possessing the right tools in compromising such networks. In guarding against these vulnerabilities, the network could be re-designed by adopting several security features as desired by the network administrator in making the network less prone to such intrusions.

3.4 Countermeasures

In re-designing the above network in fig 3.1, the following security features must be put in place as shown in the block diagram below.

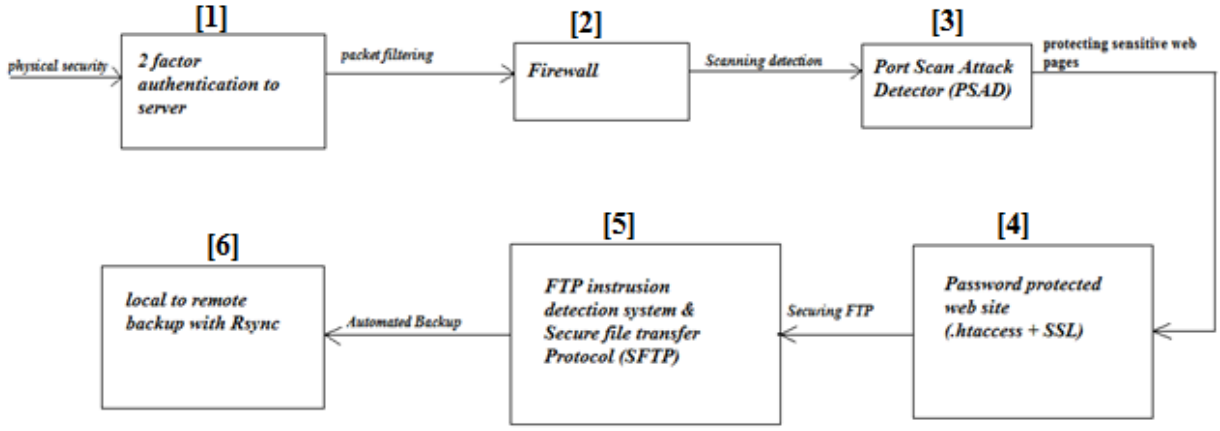

Fig 3.3 Block diagram for the Design of a Secured Network

The security features follows an organized layout in protecting the network and its servers and each layer in the security process is discussed below.

3.4.1 2 factor authentication

Physical security is provided for the servers so as to prevent unauthorized personnel's or users logging into the servers. This security is not only implemented using a password but adds a second layer for authentication which requires the presence of a specific usb drive plugged into a specific usb port as configured by the network administrator on the server. Hence in logging into the web server, a password (what the network admin knows) and usb drive (what the network admin has) is required.

3.4.2 Firewall

Immediately following the physical security implementation is the firewall. As earlier discussed, almost all of the vulnerabilities identified in the modeled network of fig 3.1 were due to the absence of a firewall. The firewall is implemented in software using the IPTABLES application available in Linux OS by default. IPTABLES is configured on the router, filtering packets based

on predefined rules set up on the input, output, and forward, pre-routing and post-routing chains. A script is created with the Linux bash shell as indicated in the appendix and scheduled to run at system/router startup using the cron scheduler. The firewall rules protect the network against attacks such as DoS, spoofing attacks, brute-force attacks etc. as commented in the script.

3.4.3 Port Scan Attack Detector (PSAD)

The PSAD is a third party software application available for installation on Linux OS. It makes use of log reports from iptables firewall to generate status messages that is sent to a network administrator when a suspicious port scan is detected. Based on settings in its configuration file, the psad sends an email alert to the network admin when a certain danger level based on the counts of packets logged is reached.

3.4.4 Intrusion Detection System for FTP Server

The IDS for the ftp server was written with the python programming language. This script/program is executed by the network administrator at startup. It reads the ftp log files at intervals and when it detects three or more failed login attempts for a particular user, it does two things; first it blocks the IP address of the pc from which the failed login is reported, and then sends an email alert to the network admin at esanboy1@gmail.com from esanboy1@yahoo.co.uk. It takes approximately 2 minutes for the mail to be received by the network administrator, after which a message is displayed on the terminal signifying a successful blocking and notification, the script then sleeps/waits for 5 mins, and continues in its while loop.

3.4.5 SSL enabled Password protected websites

In securing secret or confidential web information where access is to be granted to authorized personnel's only, a password protected web page could be implemented. This protected web page has an added layer of security – SSL (secure socket layer) to encrypt data transmitted

between the web server and client, thus making the web protocol https instead of http. With these, when authorized users enter their usernames and passwords, attackers on the same network using sniffers will be unable to make sense of any data captured since they are encrypted and are displayed as garbled alphanumeric texts.

3.4.6 Local to Remote Backup

A well designed network should provide means to back up data periodically to a backup server. Three bash shell scripts were written, where the first backs up files to a remote backup server, the second restores files from the backup server and the third lists the files on the backup server.

The backup script (first script) is scheduled with the cron utility to run at 6am and 6pm every Mondays, Wednesdays and Sundays.

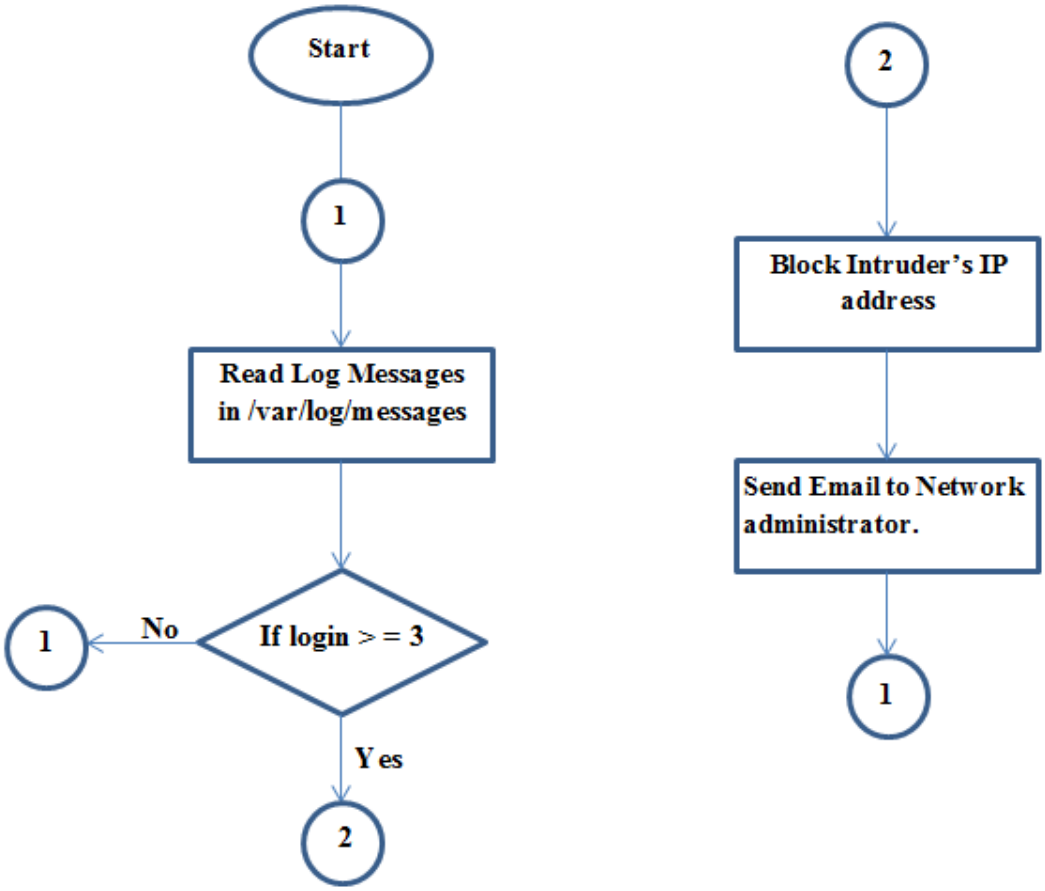

Fig 3.4 Flow Chart for FTP Intrusion Detection System

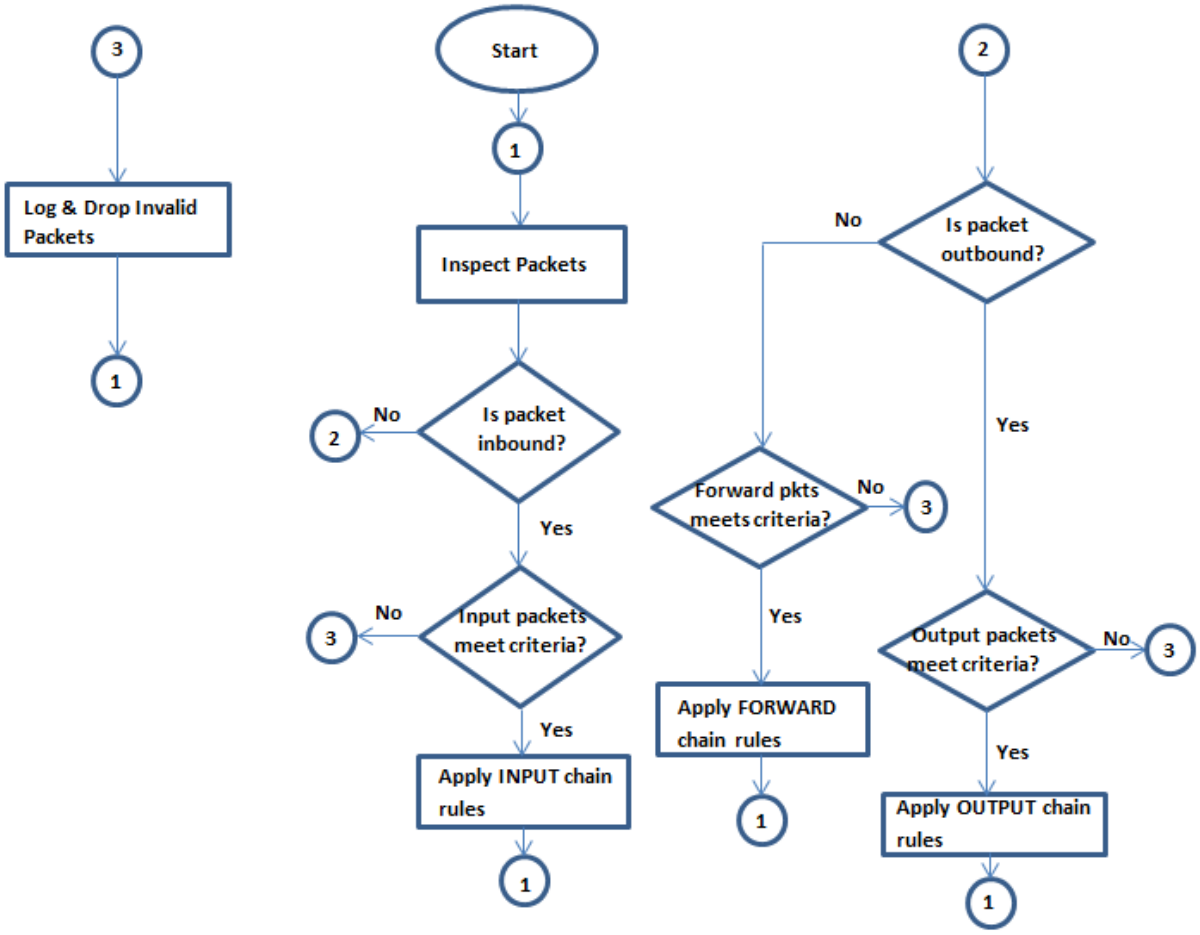

Fig 3.5 Flow Chart for a Firewall Script

3.5 Network Proxies

The review of network proxies presented in chapter two has defined another existing standard for the internet. Proxies are commonly used by LAN users who are behind a firewall, for instance if an organization restricts or blocks access to a particular web address, e.g. Facebook during office hours or pornographic sites then a user behind this firewall can bypass the this restriction by connecting to a proxy site and requesting such page from the proxy site.

A common method used by network admins in blocking access to proxy sites is gathering every proxy site known and writing firewall rules to block outgoing requests to such sites. This method has 2 major drawbacks;

- A new proxy may be created after the time the network admin gathers his proxy list, and since the firewall rules are not auto-updated, LAN users aware of such new proxy sites can still access the restricted websites.
- Heavy use of CPU resources as a result of processing of outgoing packets through the firewall against a long list of proxy sites to block.

Another method network admins use is registering with a commercial DNS operator e.g. 'safeDNS'. An account is usually created with safeDNS, after which a profile for the target organization is created. Each profile created enables the network admin configure access policies for the organization based on numerous criteria that can be selected e.g. Time of day, urls, etc. This method is less server resource intensive as it involves just reconfiguring the central gateway (router) for the organization to use the DNS server address as provided by safeDNS.

CHAPTER FOUR

RESULTS: ATTACKS & COUNTERMEASURES.

4.1 Introduction

The outcome of intrusions performed by attackers could cause all sorts of complications to a network, among which is the theft of login credentials, blackmailing of organizations to publish sensitive information if ransom isn't paid, denial of service attacks and a host of other such problems. Chapter three discussed the methodology of four intrusion methods used by attackers as gleaned from the network prototype that was set up and also provided effective countermeasures against such. This chapter presents the results of the attacks tested in chapter three and the countermeasures deployed on the network prototype.

4.2 Information Gathering Result

The graphical reconnaissance results against Landmark University are as shown in fig 4.1.

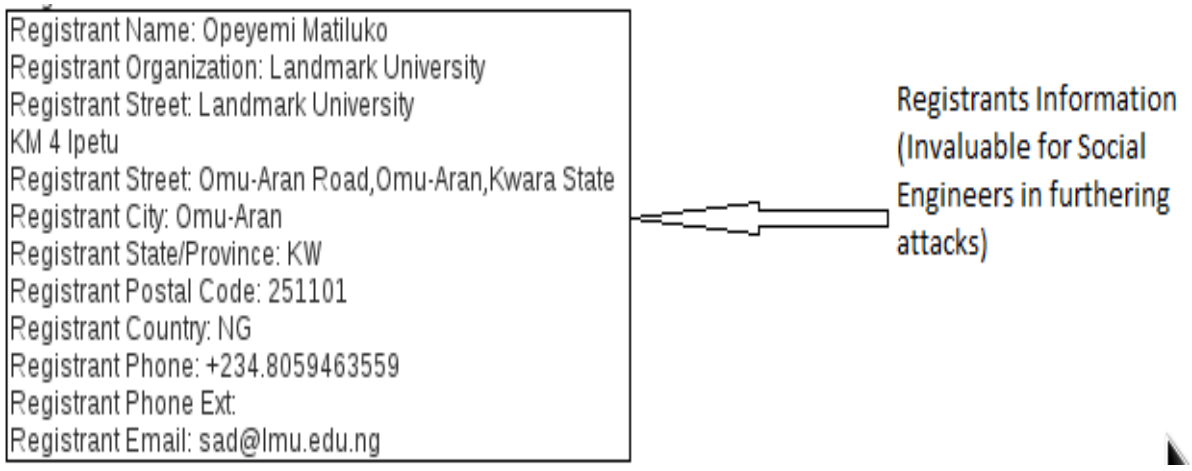

Fig 4.1 Registrant Information from www.whois.net

Fig 4.1 is an extracted view of the domain registrants' information of Landmark University.

4.3 Port Scan Results

On the virtual network environment set up in chapter 3, Fig 4.2, 4.3, and 4.4 shows the results of port scanning attacks on the virtual machines.

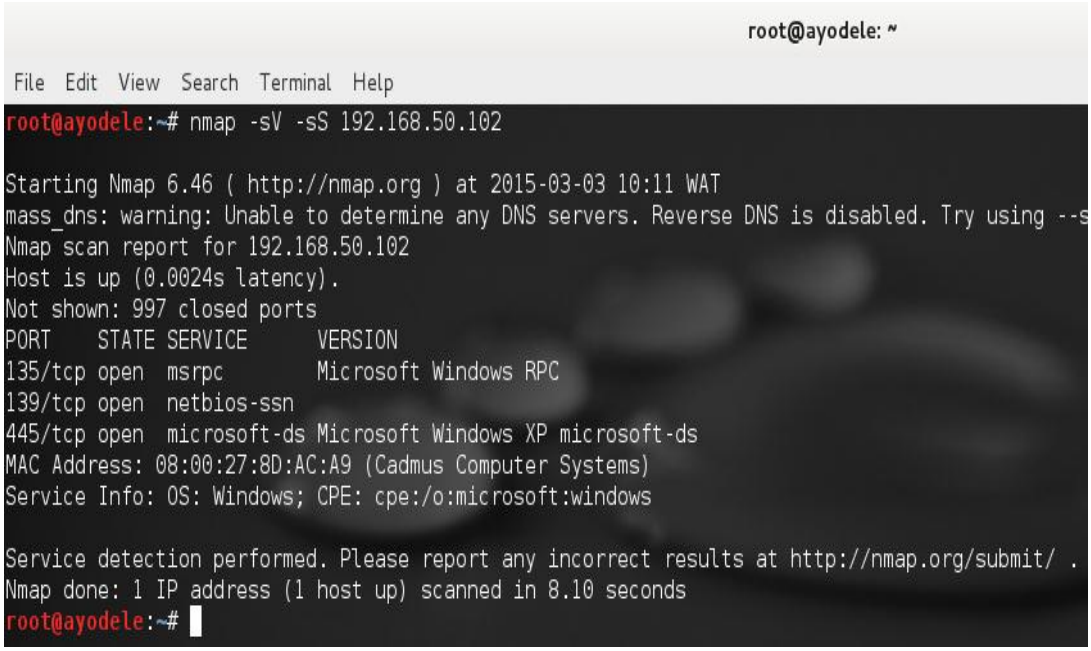

Fig 4.2 Nmap scan output on windows XP machine

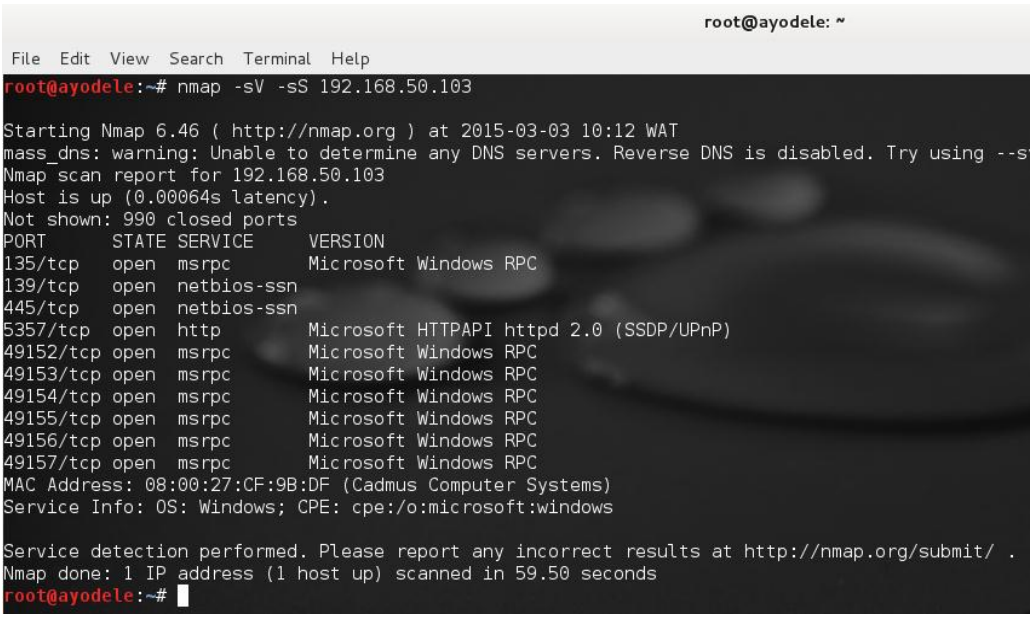

Fig 4.3 Nmap scan output on windows 7 machine

Fig 4.4 Nmap scan output on metasploitable machine

4.4 Sniffing Result

In chapter two, the shortcoming of shared networks was discussed. The results of sniffing with wireshark on Landmark Universities wireless LAN network is as shown in fig 4.5.

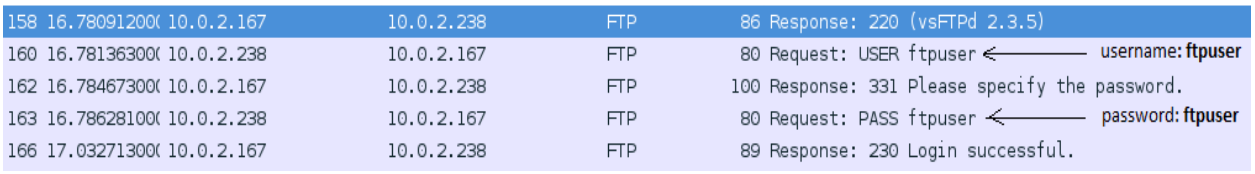

Fig 4.5 Captured ftp client username and password

4.5 ARP-Spoof Results

As earlier discussed in chapter two, hackers arp-spoof victims to intercept communication. Fig 4.6 & 4.7 shows arp-spoofing on the target and gateway respectively.

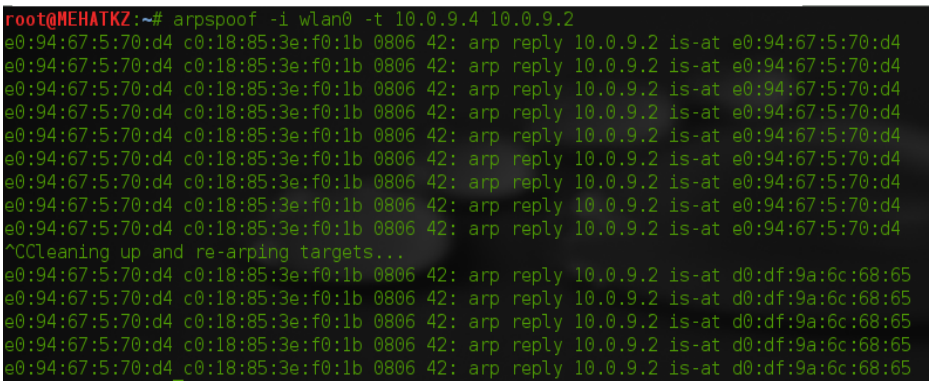

Fig 4.6 Arp spoofing the Target Computer

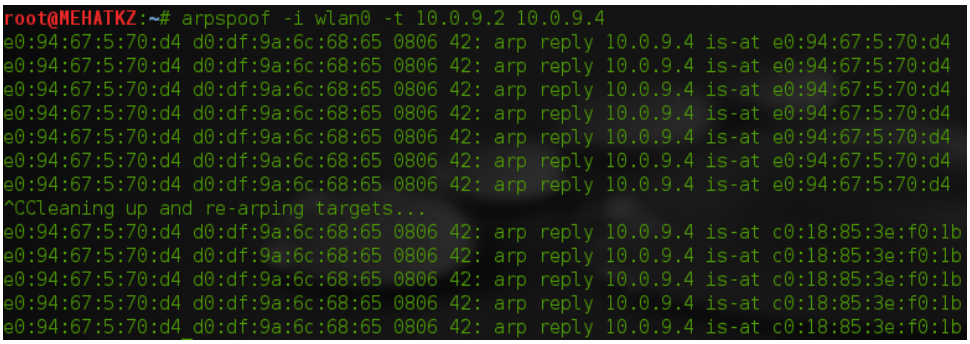

Fig 4.7 Arp spoofing the Gateway

A look at the new arp tables for the target and gateway in fig 4.8 and 4.9 shows the difference in the mac address as compared to that in the old arp table.

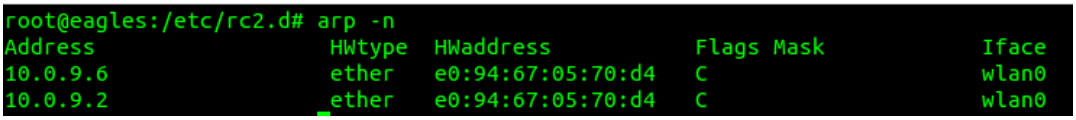

Fig 4.8 New ARP table for target

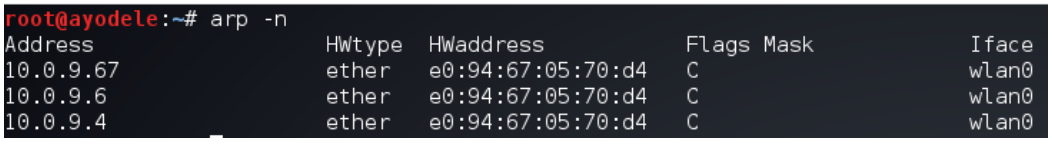

Fig 4.9 New ARP table for gateway

4.6 Countermeasures

As earlier described, the countermeasure developed for the network prototype designed in chapter three follows six (6) steps, this section shows the results for each of this step.

The first phase involves physical security to the server. Fig 4.10 shows the udev rule for usb flash drive needed for the 2 factor authentication to the Linux sever.

```
root@ayodele:/etc/udev/rules.d# cat 10-try.rules
KERNEL=="sd?1", ATTR{size}=="3939832", ATTRS{idVendor}=="1d6b", ATTRS{idProduct}=="0002", ATTRS{serial}=="0000:00:1d.0", NAME="ayoesan", RUN+="/bin/mo
unt /dev/%k /media/login/"
```

Fig 4.10 Udev rule for 2 factor authentication.

The second phase requires the implementation of the firewall rules; fig 4.11 and 4.12 shows the start and stop of the firewall script, to query the firewall status requires running the script with the 'status' argument passed to the script.

```
root@ayodele:~# ./firewall start
Loading iptables Firewall rules
Outbound Rules Successfully Implemented
Inbound Rules Successfully Implemented
Forward Rules Successfully Implemented
NAT table Rules Successfully Implemented
Port Forwarding Successfully Implemented
```

Fig 4.11 Starting the firewall

```
root@ayodele:~# ./firewall stop
About to abort Firewall...
Returning Default policies of built-in chains to ACCEPT
root@ayodele:~#
```

Fig 4.12 Aborting the Firewall.

After the firewall script is run, the third phase requires setting up the port scan attack detector; the PSAD relies on the firewall rules (especially packets that are logged) set up in the preceding phase. Fig 4.13 and 4.14 shows the status of the PSAD before and after an nmap scan is performed on the server.

```
root@ayodele:/var/mail# service psad status
Status of Port Scan Attack Detector:
[+] psadwatchd (pid: 7381)  %CPU: 0.0  %MEM: 0.0
    Running since: Wed Apr  8 01:18:01 2015

[+] psad (pid: 7379)  %CPU: 0.1  %MEM: 0.3
    Running since: Wed Apr  8 01:18:01 2015
    Command line arguments: [none specified]
    Alert email address(es): moyo@eagles.lab

[+] Version: psad v2.2

[+] Top 50 signature matches:
        [NONE]

[+] Top 25 attackers:
        [NONE]

[+] Top 20 scanned ports:
        [NONE]

[+] iptables log prefix counters:
        [NONE]

    Total packet counters:

[+] IP Status Detail:
        [NONE]

    Total scan sources: 0
    Total scan destinations: 0

[+] These results are available in: /var/log/psad/status.out
```

Fig 4.13 Initial status of psad

```
[+] Top 25 attackers:
     10.0.9.4        DL: 4, Packets: 2024, Sig count: 76
     127.0.0.1       DL: 2, Packets: 63, Sig count: 0, (local IP)

[+] Top 20 scanned ports:
     tcp 631   13 packets
     tcp 80    4 packets
     tcp 5666  2 packets
     tcp 8009  2 packets
     tcp 8300  2 packets
     tcp 1169  2 packets
     tcp 5679  2 packets
     tcp 1049  2 packets
     tcp 32    2 packets
     tcp 5030  2 packets
     tcp 90    2 packets
     tcp 443   2 packets
     tcp 1801  2 packets
     tcp 6006  2 packets
     tcp 8010  2 packets
     tcp 9091  2 packets
     tcp 15000 2 packets
     tcp 5560  2 packets
     tcp 4449  2 packets
     tcp 1434  2 packets

     udp 67    49 packets
     udp 512   26 packets
     udp 53    26 packets
     udp 68    25 packets

[+] iptables log prefix counters:
     "Dropped by Default:": 2136
```

Fig 4.14 New PSAD status

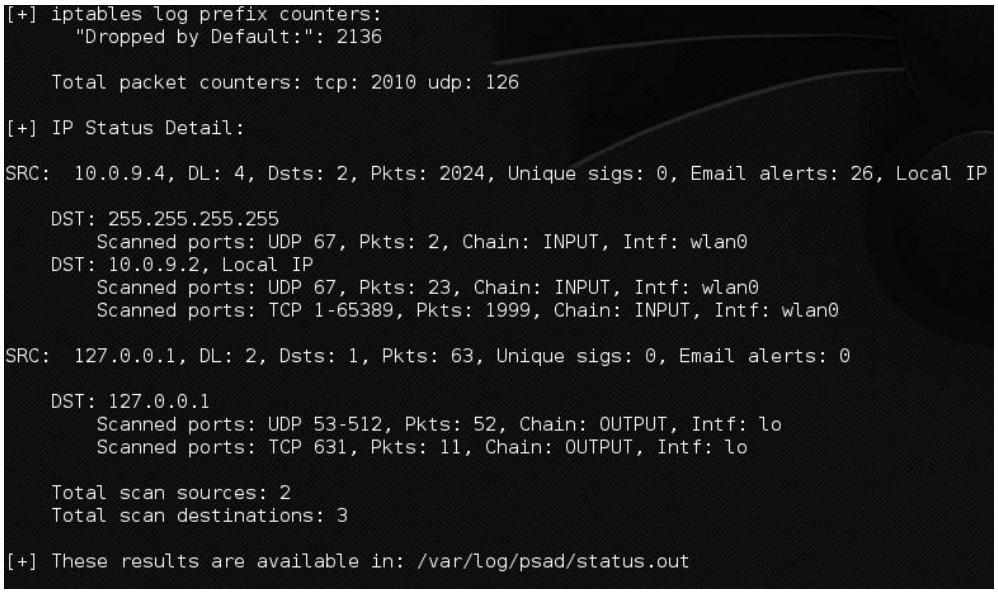

Fig 4.15 Cont. of New PSAD status

The fourth phase provides security for confidential web pages stored on the web server. The web server implements ssl and .htaccess to encrypt web data and provides a password protection mechanism for the web pages. Fig 4.16 and 4.17 shows the key and certificate generated for our web server.

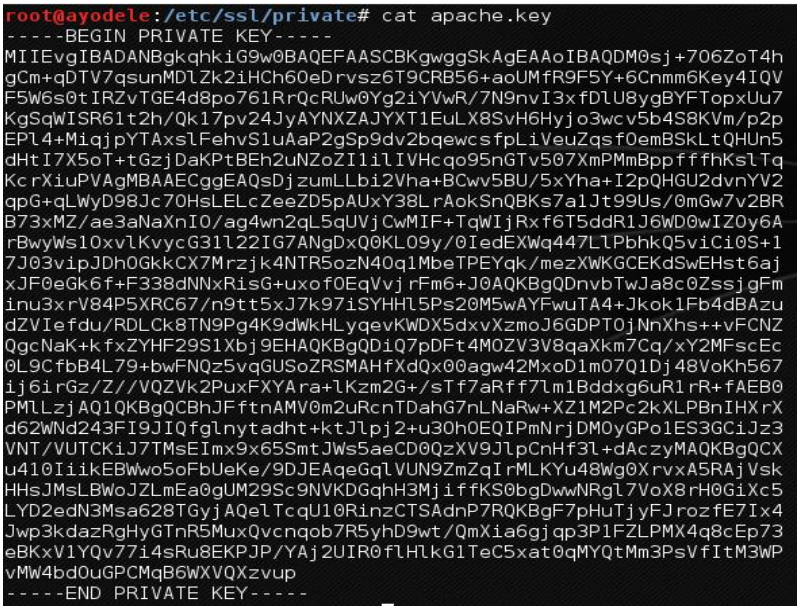

Fig 4.16 SSL key for apache web server

Fig 4.17 SSL certificate for apache

Following the ssl key and self-signed certificate setup, the password utility for apache2 is setup and configured to protect the secret webpage at /var/www/site1/. Fig 4.18 shows the result of entering the url (https://www.eagles.lab) of the webpage on a browser.

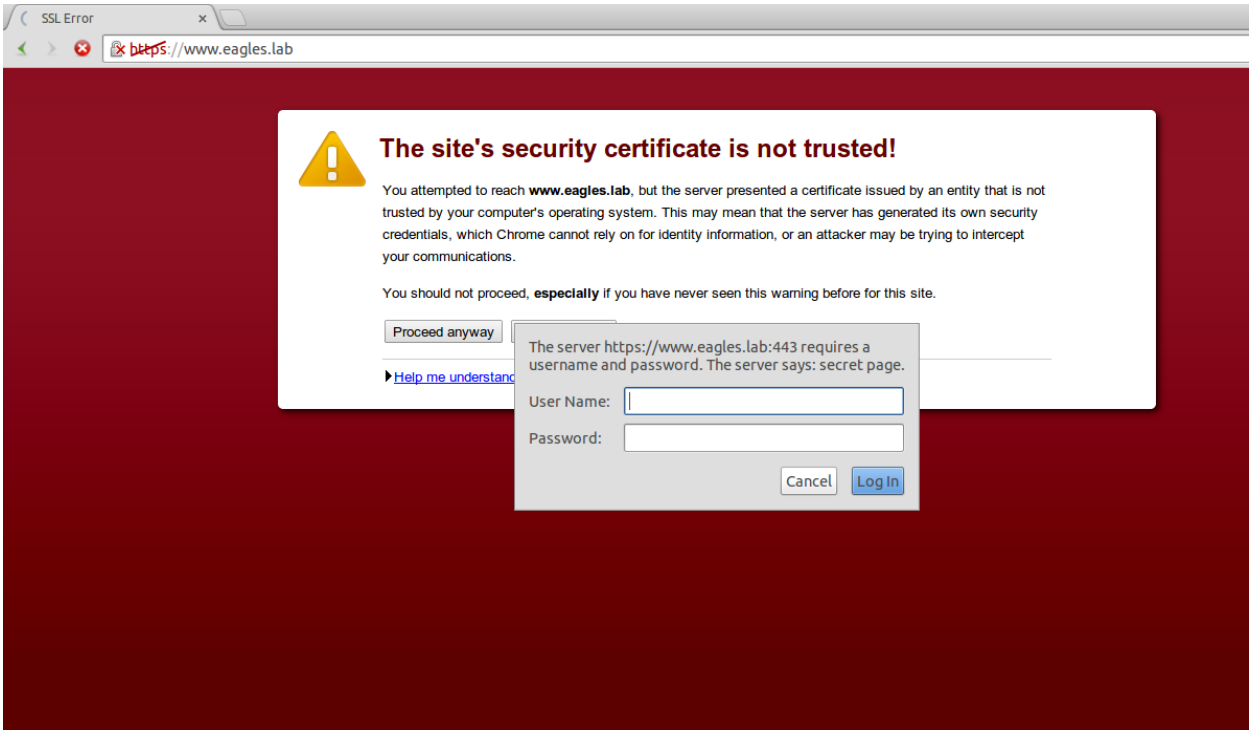

Fig 4.18 SSL enabled PPS

Fig 4.19 shows the content of the secret webpage only after the correct usernames and password are entered.

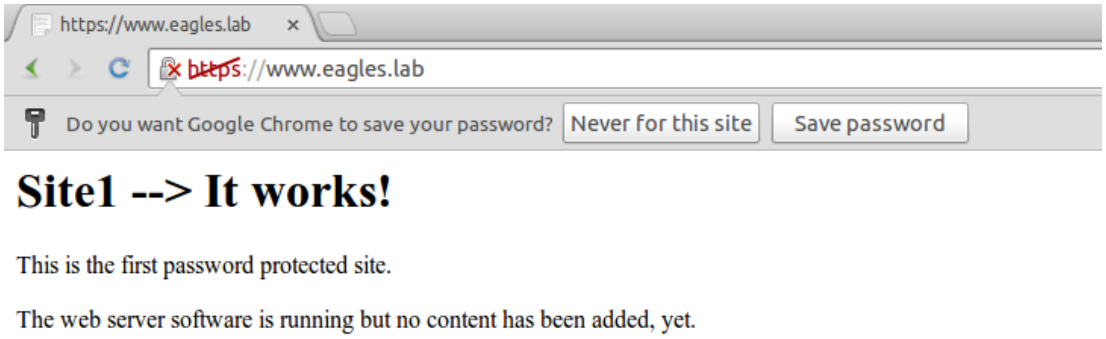

Fig 4.19 Content of password protected site (PPS)

The fifth phase secures the FTP server. An ftp intrusion detection system is being run to watch for wrong username/password of more than 3 times. Fig 4.20 shows the execution of the IDS python script on the command line.

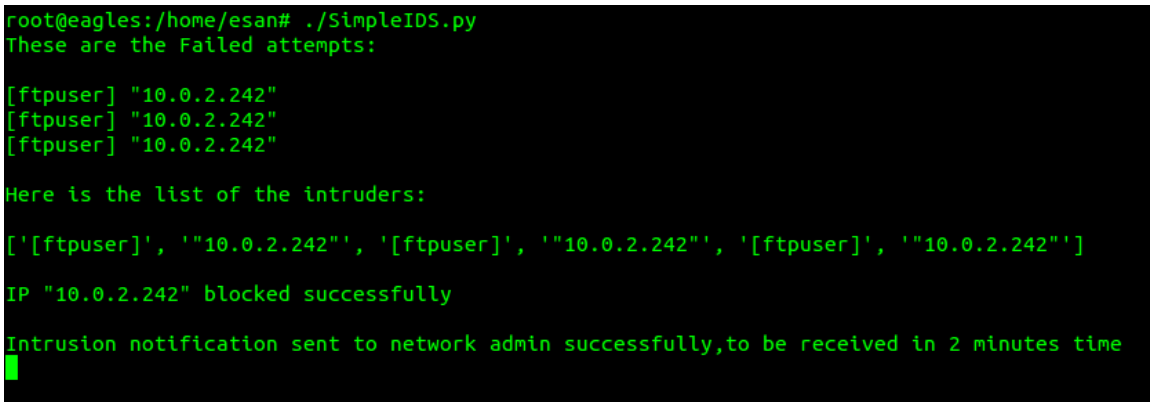

Fig 4.20 Execution of SimpleIDS.py on FTP server

The script keeps checking the ftp logs and detects users and IP address of intruders and a notification is sent to the network admin. It takes approximately 2 minutes for the mail to get to the network admin (esanboy1@gmail.com). Fig 4.21 shows the email notification sent to the network administrator.

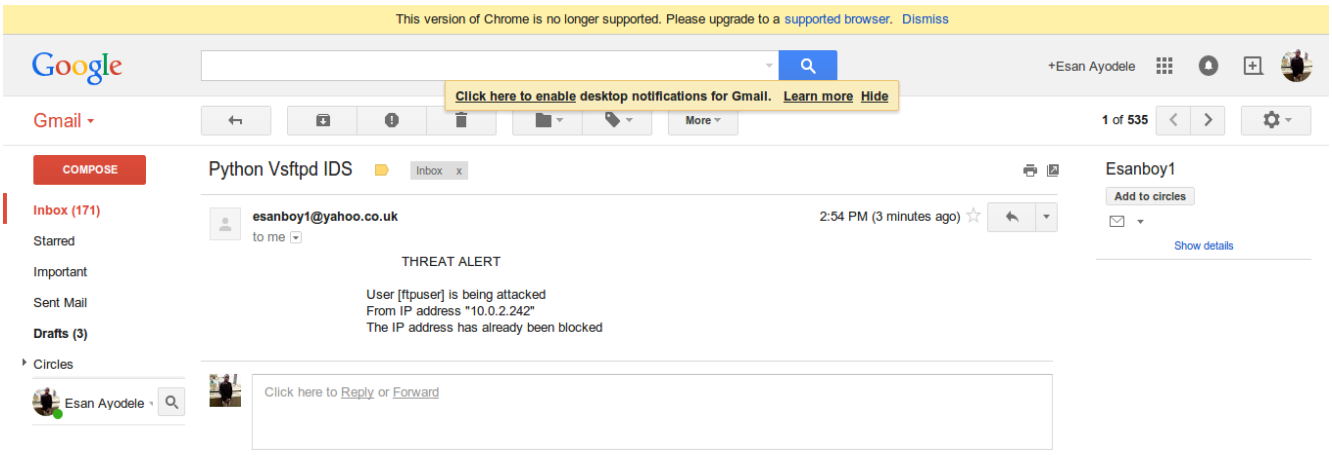

Fig 4.21 Email content sent to network admin

The sixth and final phase is an automated scheduled backup. In case of data loss or data theft, sensitive data should be periodically backed up to a remote server. The rsync utility is used as discussed in chapter 3. Fig 4.22 shows the result of the execution of the backup script.

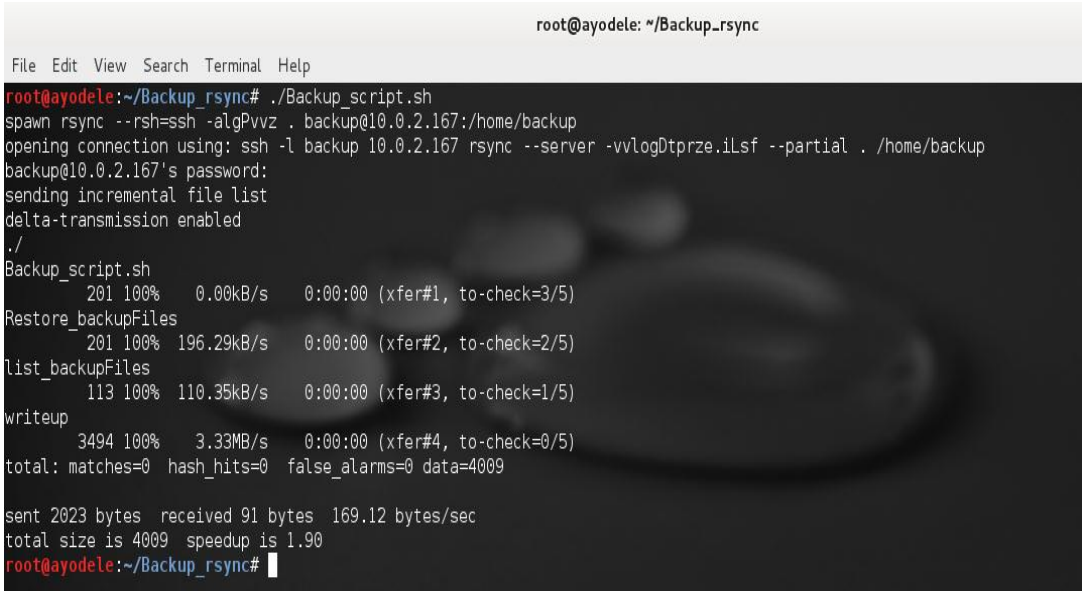

Fig 4.22 Running the Backup_script

Fig 4.23 shows the result of executing the script to list the files on the backup server.

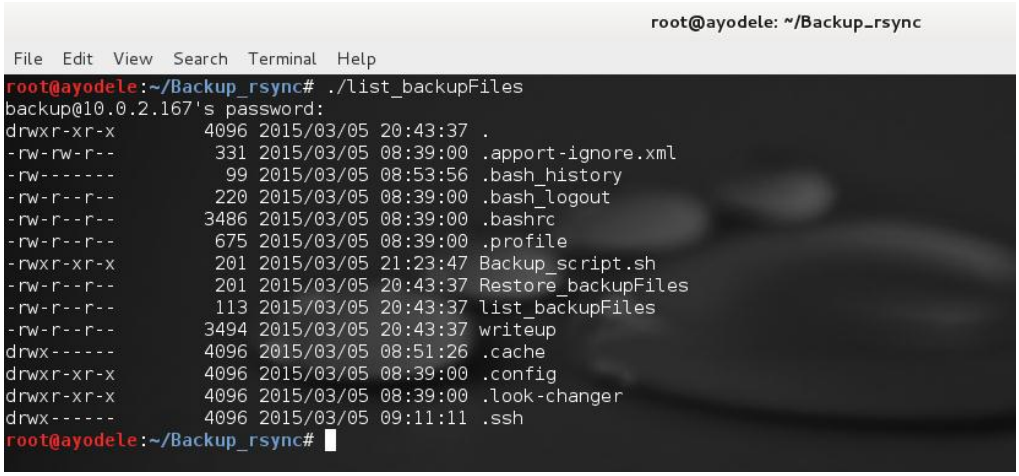

Fig 4.23 Running the list_backupFile script

Fig 4.24 shows the result of executing the script to restore file(s) from the remote backup server to the local server.

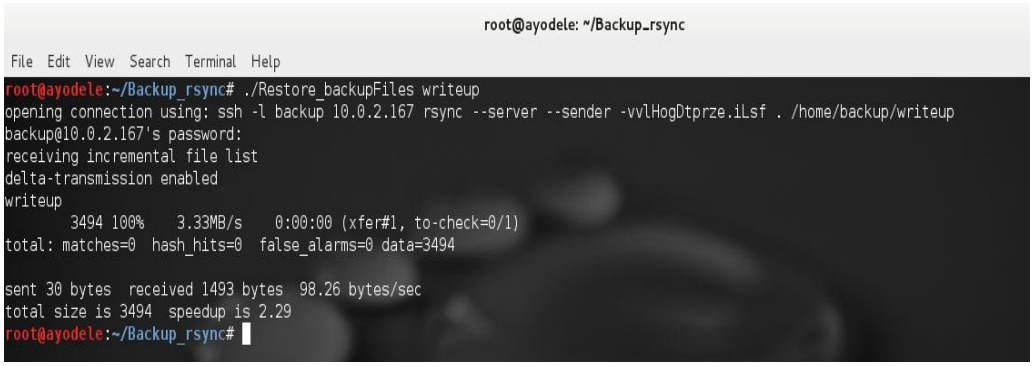

Fig 4.24 Running the Restore_backupFiles script

Depending on the applications/services on the server, more security phase could be added to the above phase. It is also important to note that although these security phases could strengthen the network, latest updates and patches on applications/services (especially security bug fixes) should be downloaded and installed.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

This study reviewed a couple of intrusion methods through which persons with malicious intents compromise personal data or intercept the communication of users on a network. It also reviewed countermeasures developed by different individuals in mitigating attacks aimed at networks. The stated objectives for this project mainly focused on developing countermeasures against these intrusions. These objectives were realized as a network security model comprising a six tier layer was implemented to protect servers on a network. These six tiers are as listed below:

1) The Two factor authentication,
2) The Firewall setup,
3) The Port Scan Attack Detector,
4) The Password protected website with SSL enabled,
5) The Intrusion detection system for FTP servers and,
6) A Local to Remote backup with Rsync

The network security model utilized a host of utilities many of which are Linux proprietary and others which works on other operating system software e.g. Windows. The Two factor authentication uses the UDEV daemon for hardware management of the USB device, the firewall uses the IPTABLES Linux firewall, *PSAD* is used for port scan detection, SSL and HT-ACCESS was used to create an SSL enabled passworded webpage, an IDS designed for ftp security was created using PYTHON programming language, while the backup uses a combination of

RSYNC, BASH shell, as well as the CRON utility in three short scripts for backup, listing and restoration of files to/from the remote backup server.

5.2 Conclusion

The internet has the potential to transform societies, with benefits for education, health, agriculture, transportation, governance, culture, business, and economies – in fact the benefits of the internet can reach into every aspect of modern human society. However a certain threshold of internet security – including physical security, network devices security, awareness, firewalls deployment, web security etc. – is required before organizations and industries are able to realize confidentiality in the data they harbor. Thus any network design and implementation without adequate plan and attention given to the security and protection of network devices, data stored on servers and even security to its clients would be considered fatal and futile.

5.3 Recommendation

This project only looked at a few of intrusions methods, even though a myriad of them exist. Therefore a thorough investigation should be made to understand other intrusion methods used including those targeted at compromising websites and database of organizations. Also the countermeasures were only tested on a small network, hence proper testing could be made on larger networks and monitored to determine if it scales well. The scripts for the firewall could further be improved to accommodate more firewall rules to detect and prevent forged or invalid packets and that of the Intrusion detection system could be developed to secure other network applications and include a database to store the network information (ip-address) to quarantine such attacking ip addresses from accessing the network.

REFERENCES

[1] bigplanetusa. "Understanding internet security". http://www.bigplanetusa.com/library/bp/pdf/bpis_understanding_security.pdf.

[2] Forbes. "The Top 5 Most Brutal Cyber Attacks Of 2014 So Far". http://www.forbes.com/sites/jaymcgregor/2014/07/28/the-top-5-most-brutal-cyber-attacks-of-2014-so-far/.

[3] Syngress. "Network Security Basics". http://scitechconnect.elsevier.com/wp-content/uploads/2013/09/Network-Security-Basics.pdf.

[4] Peter A. Akhihiero ESQ. "Internet Security". Basic Computer Training Mikon Institute of Information Technology, pp. 5, May 2006

[5] hackertoolkit. "The Hackers Tool kit". http://www.hackertoolkit.com/

[6] tripwire. "Top Five Hacker Tools Every CISO Should Understand". http://www.tripwire.com/state-of-security/security-data-protection/top-five-hacker-tools-every-ciso-should-understand/

[7] coveros. "An Introduction to Kali Linux". https://www.coveros.com/an-introduction-to-kali-linux/

[8] edge-security. "A fresh new look into Information Gathering". http://www.edge-security.com/docs/OWASP-Christian_Martorella-InformationGathering.pdf.

[9] infosecinstitute. "Network Intelligence Gathering" http://resources.infosecinstitute.com/network-intelligence-gathering/

[10] cisco. "Advanced Attacks using tcp/ip for scanning". https://supportforums.cisco.com/blog/153536

[11] Monowar H Bhuyan, D K Bhattacharyya and J K Kalita. “Surveying Port Scans and Their Detection Methodologies”, A proceeding of the the Computer Journal, August 23 2010, pp. 1-2

[12] valencynetworks. “Cyber security Network sniffing”. http://www.valencynetworks.com/articles/cyber-security-attacks-network-sniffing.html

[13] Sumit-Dhar. “Sniffers Basics and Detection”. http://www.just.edu.jo/~tawalbeh/nyit/incs745/presentations/Sniffers.pdf

[14] paloaltonetworks. “Denial of service attack - prevent Dos with palo alto networks”. https://www.paloaltonetworks.com/resources/learning-center/what-is-a-denial-of-service-attack-dos.html

[15] cert. “Denial of service (published 1997)”. https://www.cert.org/information-for/denial_of_service.cfm

[16] Stephen Fewer, “ARP Poisoning - An investigation into spoofing the Address Resolution Protocol” Harmony Security Research and Consultancy, pp. 2-3, March 2007

[17] whatis. “What is two-factor authentication (2FA)?”. http://www.WhatIs.com/

[18] Wikipedia. “Two-step verification”. https://en.wikipedia.org/wiki/Two-step_verification

[19] brown . “Firewalls, Tunnels, and Network Intrusion Detection”. http://cs.brown.edu/cgc/net.secbook/se01/handouts/Ch06-Firewalls.pdf”

[20] wiley. “Firewalls and Virtual Private Networks”. http://www.wiley.com/legacy/compbooks/press/0471348201_09.pdf

[21] cipherdyne. “psad: Intrusion Detection and Log Analysis with iptables”. https://cipherdyne.org/psad/

[22] paloaltonetworks. “what is intrusion detection system”. https://www.paloaltonetworks.com/network-infrastructure/what-is-intrusion-detection-system-pdf.html

[23] dtic. “Intrusion Detection System”. http://iac.dtic.mil/csiac/download/intrusion_detection.pdf

[24] digicert. “What is SSL and what are SSL certificates”. https://www.digicert.com/ssl.htm

[25] instantssl. “What’s HTTPS”?. https://www.instantssl.com/ssl-certificateproducts/https.html

[26] admin-magazine. “Incremental backup on linux”. http://www.admin-magazine.com/Articles/Using-rsync-for-Backups

[27] marksanborn. “Use rsync for daily, weekly and full monthly backups”. http://www.marksanborn.net/howto/use-rsync-for-daily-weekly-and-full-monthly-backups/

[28] McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T., and Christensen, M. (2004), “PortVis: A tool for port-based detection of security events.”, A Proceedings of VizSEC/DMSEC’04, Washington, DC, USA, October 29, pp. 73–81. ACM

[29] Mehiar Dabbagh, Ali J. Ghandour, Kassem Fawaz, Wassim El Hajj, and Hazem Hajj (2011), “Slow Port Scanning Detection”,A proceeding of the 7th InternationalConferenceonInformationAssuranceandSecurity(IAS),2011, pp. 228-229.

[30] Chen, J.-j. and Cheng, X.-j. (2009), “A novel fast port scan method using partheno-genetic algorithm.”, Proceedings of ICCSIT’09, Los Alamitos, CA, USA, August, 8-11, pp. 219–222. IEEE Computer Society

[31] Network Traffic analysis and Intrusion detection using packet sniffer_steven.pdf

[32] Handley, M. and Rescorla, E. (2006). Internet Denial of Service Considerations. Available at: http://tools.ietf.org/html/draft-iab-dos-05. (Date of access: October 31, 2006)

[33] Ai-zeng Qian, "The Automatic Prevention and Control Research of ARP Deception and Implementation", 2009 WRI World Congress on Computer Science and Information Engineering, , 2(1), pp. 555 558, April 2009

[34] Boughrara, A.; Mammar, S., "Implementation of a SNORT's output Plug-In in reaction to ARP Spoofing's attack", 2012 6th International Conference on Sciences of Electronics Technologies of Information and Telecommunications (SETIT), pp.643,647, 21-24 March 2012

[35] Danilo Bruschi, Alberto Ornaghi, Emilia Rosti , “S-ARP: a Secure Adderess Resolution Protocol”, 19th Annual Computer Security Applications Conference, 2003, www.acsac.org/2003/papers/111.pdf

APPENDIX A

Firewall Script

```
#!/bin/bash
IPTABLES=/sbin/iptables
test -x $IPTABLES || exit 5 #this checks if iptables is present and whether or not its executable
case "$1" in
start)
echo "Loading iptables Firewall rules"
#Load kernel modules for iptables firewall
modprobe ip_tables
modprobe ip_conntrack_ftp
modprobe iptable_nat
modprobe ip_nat_ftp
modprobe ipt_MASQUERADE
modprobe nf_conntrack_ipv4
#Flush all rules from default chains, and delete any custom chain
$IPTABLES -F
$IPTABLES -X
########################OUTPUT chain########################
#Default policy for output chain
$IPTABLES -P OUTPUT DROP
#Set up state connection tracking
$IPTABLES -A OUTPUT -m state --state INVALID -j LOG --log-prefix "INVALID: " --log-ip-options --log-tcp-options
$IPTABLES -A OUTPUT -m state --state INVALID -j DROP
$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#Setup outbound rules
$IPTABLES -A OUTPUT -s 10.0.9.0/24 -p tcp -m multiport --dports 80,443,22,25 -m state --state NEW --syn -j ACCEPT
$IPTABLES -A OUTPUT -s 10.0.9.0/24 -p icmp --icmp-type echo-request -j ACCEPT
```

27. $IPTABLES -A OUTPUT -s 10.0.9.0/24 -p udp --dport 53 -m state --state NEW -j ACCEPT
28. $IPTABLES -A OUTPUT -s 10.0.9.0/24 -p tcp --dport 53 -m state --state NEW --syn -j ACCEPT
29. $IPTABLES -A OUTPUT -s 10.0.9.0/24 -p tcp -m state --state NEW --syn -m time --timestart 21:00 --timestop 05:00 --utc -j DROP
30. #Log everything else not accepted above
31. $IPTABLES -A OUTPUT -j LOG --log-prefix "Dropped by Default: "
32. echo "Outbound Rules Successfully Implemented"
33. ########################INPUT chain########################
34. #Default policy for input chain
35. $IPTABLES -P INPUT DROP
36. #Set up state connection tracking
37. $IPTABLES -A INPUT -m state --state INVALID -j LOG --log-prefix "INVALID: " --log-ip-options --log-tcp-options
38. $IPTABLES -A INPUT -m state --state INVALID -j DROP
39. $IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
40. #Allow local interface
41. $IPTABLES -A INPUT -i lo -j ACCEPT
42. #Allow only 3 SSH connections per client host (Mitigate brute-force attacks against SSH server)
43. $IPTABLES -A INPUT -i wlan0 -p tcp --dport 22 -m state --state NEW --syn -m connlimit --connlimit-above 3 -j DROP
44. #Allow only 10 conections in within 100 secs to ports 80/443 (Mitigates DoS attacks)
45. #The latch that sets the rule
46. $IPTABLES -A INPUT -i wlan0 -p tcp -m state --state NEW --syn -m multiport --dports 80,443 -m recent --set
47. #The trip that enforces the rule
48. $IPTABLES -A INPUT -i wlan0 -p tcp -m state --state NEW --syn -m multiport --dports 80,443 -m recent --update --seconds 100 --hitcount 10 -j DROP
49. #Allow incoming DNS connections on UDP and TCP

50. $IPTABLES -A INPUT -p udp -m state --state NEW --dport 53 -j ACCEPT
51. $IPTABLES -A INPUT -p tcp -m state --state NEW --syn --dport 53 -j ACCEPT
52. #Allow incoming connection to mail server (smtp)
53. $IPTABLES -A INPUT -i wlan0 -p tcp -m state --state NEW --syn --dport 25 -j ACCEPT
54. #Allow icmp ping requests,but only 10 packets per minute, with limit-burst of 50,meaning limit/minute rule will be enforced only #after total number of connection within a minute reaches the limit-burst level i.e 50
55. $IPTABLES -A INPUT -i wlan0 -p icmp --icmp-type echo-request -m limit --limit 10/m --limit-burst 50 -j ACCEPT
56. #Log everything else not accepted above
57. $IPTABLES -A INPUT -j LOG --log-prefix "Dropped by Default: "
58. echo "Inbound Rules Successfully Implemented"
59. ##########################FORWARD chain##########################
60. #Default Forward Policy
61. $IPTABLES -P FORWARD DROP
62. #Set up state connection tracking
63. $IPTABLES -A FORWARD -m state --state INVALID -j LOG --log-prefix "INVALID: " --log-ip-options --log-tcp-options
64. $IPTABLES -A FORWARD -m state --state INVALID -j DROP
65. $IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
66. #Allow DNS connections
67. $IPTABLES -A FORWARD -p udp -m state --state NEW --dport 53 -j ACCEPT
68. $IPTABLES -A FORWARD -p tcp -m state --state NEW --syn --dport 53 -j ACCEPT
69. #Forward http,https,smtp,ssh connections from eth1 to wlan0 and vice-versa
70. $IPTABLES -A FORWARD -i eth1 -o wlan0 -p tcp -m state --state NEW --syn -m multiport --dports 80,443,25,22 -j ACCEPT
71. $IPTABLES -A FORWARD -i wlan0 -o eth1 -p tcp -m state --state NEW --syn -m multiport --dports 80,443,25,22 -j ACCEPT
72. #Log everything else not forwarded above
73. $IPTABLES -A FORWARD -j LOG --log-prefix "Dropped by Default: "

74. echo "Forward Rules Successfully Implemented"
75. ########################NAT TABLE (PREROUTING AND POSTROUTING CHAINS)########################
76. #PREROUTING CHAIN
77. #Redirect all http traffic to Internal IP 10.0.5.12
78. $IPTABLES -t nat -A PREROUTING -i wlan0 -p tcp --dport 80 -j DNAT --to-destination 10.0.9.12:80
79. #Redirect all https traffic to Internal IP 10.0.5.12
80. $IPTABLES -t nat -A PREROUTING -i wlan0 -p tcp --dport 443 -j DNAT --to-destination 10.0.9.12:443
81. #Redirect all SSH connection to Internal IP 10.0.5.13
82. $IPTABLES -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j DNAT --to-destination 10.0.9.13:22
83. #Redirect all SMTP connection to Internal IP 10.0.5.13
84. $IPTABLES -t nat -A PREROUTING -i wlan0 -p tcp --dport 25 -j DNAT --to-destination 10.0.9.13:25
85. #Redirect all DNS requests to Internal IP 10.0.5.14
86. $IPTABLES -t nat -A PREROUTING -i wlan0 -p tcp --dport 53 -j DNAT --to-destination 10.0.5.14:53
87. $IPTABLES -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j DNAT --to-destination 10.0.5.14:53
88. $IPTABLES -t nat -A POSTROUTING -s 10.0.9.0/24 -o wlan0 -j MASQUERADE
89. echo "NAT table Rules Successfully Implemented"
90. ###############Set Up Port Forwarding#################
91. echo "1" > /proc/sys/net/ipv4/ip_forward
92. echo "Port Forwarding Successfully Implemented"
93. ;;
94. stop)
95. echo "About to abort Firewall..."
96. #unload all firewall rules, leaving default-drop policies
97. $IPTABLES -F

```
98. $IPTABLES -X
99. $IPTABLES -t nat -F
100.        $IPTABLES -t nat -X
101.        echo "Returning Default policies of built-in chains to ACCEPT"
102.        $IPTABLES -P INPUT ACCEPT
103.        $IPTABLES -P OUTPUT ACCEPT
104.        $IPTABLES -P FORWARD ACCEPT
105.        ;;
106.        status)
107.        echo "Querying iptables status (via iptables --list)..."
108.        echo "[+]  Filter table status  [+]"
109.        $IPTABLES -nvL --line-numbers
110.        echo "[+]  Nat table status  [+]"
111.        $IPTABLES -t nat -nvL --line-numbers
112.        ;;
113.        *)
114.        echo "Usage: $0 {start|stop|status}"
115.        exit 1
116.        ;;
117.        esac
```

APPENDIX B

Local to Remote Backup Script

BackupFile Script

```
1. #!/usr/bin/expect
2. set timeout 20
3. spawn rsync --rsh=ssh -algPvvz . backup@10.0.2.167:/home/backup
4. expect "backup@10.0.2.167's password: " { send "backup\r" }
5. expect "root@ayodele:~/Backup_rsync#"
```

RestoreFile Script

1. #!/bin/bash
2. #Restoring lost or damaged files
3. for file in "$@"
4. do
    a. rsync -aHPvvz backup@10.0.2.167:/home/backup/${file} ./${file}
5. done

ListFiles Script

1. #!/bin/bash
2. #To list all files on your backup server using rsync
3. rsync backup@10.0.2.167:/home/backup/ | more

APPENDIX C

FTP IDS script

1. #!/usr/bin/python
2. import subprocess
3. import smtplib
4. import time
5. from email.MIMEText import MIMEText
6. intrusion = "cat /var/log/vsftpd.log | grep FAIL | awk '{print $11,$12}'"
7. freq = intrusion+"| wc -l"
8. while True:

```
exec1=
subprocess.Popen(freq,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIP
E)
res1 = exec1.communicate()
num_freq = int(res1[0].strip())
if (num_freq >= 3):
```

```
exec2=
subprocess.Popen(intrusion,shell=True,stdout=subprocess.PIPE,stderr=su
bprocess.PIPE)
res2 = exec2.communicate()
a = res2[0].strip()
print "These are the Failed attempts\n"
print a+'\n'
b = a.split()
print str(b)+'\n'
if (b[0] == b[2] == b[4]):
        Intruder = b[0]
else:
        continue
if (b[1] == b[3] == b[5]):
        Intru_ip = b[1]
else:
        continue

block_ip = "iptables -A INPUT -s "+Intru_ip+" -p tcp --dport 21 -j
REJECT"
block_exec=
subprocess.Popen(block_ip,shell=True,stdout=subprocess.PIPE,stderr=sub
process.PIPE)
block_res = block_exec.communicate()
if block_res[1] == '':
        print 'IP '+Intru_ip+' blocked successfully\n'
else:
        print 'IP '+Intru_ip+' couldnt be blocked\n'
Yahoo_smtp = "smtp.mail.yahoo.com"
Yahoo_smtp_port = 465
content = '''
```

```
                THREAT ALERT
        User '"+Intruder+"' is being attacked
        From IP address '"+Intru_ip+"'
        The IP address has already been blocked"'
Text_subtype = "plain"
username = str('esanboy1@yahoo.co.uk')
password = str('xxxxxxxxxxx')
msg = MIMEText(content, Text_subtype)
msg["Subject"] = 'Python Vsftpd IDS'
msg["From"] = 'IDS_watch'
msg["To"] = 'esanboy1@gmail.com'
try:
        s = smtplib.SMTP_SSL(Yahoo_smtp,Yahoo_smtp_port)
        s.login(user=username,password=password)
        s.sendmail(username,msg["To"], msg.as_string())
        s.quit()
        print 'Intrusion notification sent to network admin successfully,to
        be received in 2 minutes time'
        time.sleep (300) # sleep for 5 mins
        continue # run the while loop again
except:
        print 'can't send mail'
        break
```