SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

NETWORK SECURITY: ATTACKS AND DEFENSE MECHANISM BY DESIGNING AN INTELLIGENT FIREWALL AGENT

M.Sc. THESIS

Hafuswa NAKATO

Department	**:**	**COMPUTER AND INFORMATION ENGINEERING**
Supervisor	**:**	**Prof. Dr. İsmail Hakkı CEDİMOĞLU**

June 2016

SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

NETWORK SECURITY: ATTACKS AND DEFENSE MECHANISM BY DESIGNING AN INTELLIGENT FIREWALL AGENT

M.Sc. THESIS

Hafuswa NAKATO

Department : COMPUTER AND INFORMATION ENGINEERING

Supervisor : Prof. Dr. İsmail Hakkı CEDİMOĞLU

This thesis has been accepted unanimously / with majority of votes by the examination committee on the 13.06.2016

Prof. Dr.
Şeref SAĞIROĞLU

Head of Jury

Prof. Dr.
İsmail Hakkı CEDİMOĞLU

Jury Member

Prof. Dr.
Orhan TORKUL

Jury Member

DECLARATION

I declare that all the data in this thesis was obtained by myself in accordance with academic rules, all visual and written information and results were presented in accordance with academic and ethical rules, there is no distortion in the presented data, in case of utilizing other people's work it were referenced properly according to scientific norms, the data presented in this thesis has not been used in any other thesis in this university or in any other university.

Hafuswa NAKATO
13.6.2016

ACKNOWLEDGEMENT

In The Name of **ALLAH**, the Most Beneficial and Merciful. Am greatly very thankful to all those people who have helped me in giving us support throughout performing our thesis. Firstly, i would like to thank my research supervisors: Dr Prof Ismail Hakki Cedimoğlu for all his academic support and advice he rendeered to me during my study. Not forgetting Prof Dr Orhan Torkul for inspiring my interest in the subject, and for patiently and painstakingly increasing my knowledge of it. Not forgetting all my teachers who helped me during m grduate study; Raşit Cesur, Assoc. Prof. Dr. Murat İskefiyeli, Prof. Dr. Celal Çeken, Assoc. Prof. Dr. İbrahim Özçelik, Prof.Dr İbrahim Çil and Assoc. Prof. Dr Seçkin Arı. Great thanks also goes to my external supervisor Prof.Dr.Şeref Sağıroğlu for giving in his time to come and be my external supervisor.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my parents, brothers and sisters for always encouraging and supporting me throughout my whole study. I would not have finished this thesis. The support moral support and assistance from my friends also helped me; Mr. Kasule Abdallah, Mrs. Belaynesh Chekol. Lastly, the moral support given by my boyfriend Swalehe Masoud throughout the writing of my report and without his love, encouragement and editing assistance of the report was so critical to the completion of this work.

In conclusıon, i recognize that this research would not have been possible without the assistance from Sakarya University-Department of Computer Engineering.

TABLE OF CONTENTS

CHAPTER 3.

CHAPTER 4.

LIST OF SYMBOLS AND ABBREVIATIONS

ACK	: Acknowledgement
ACS	: Acess Control System
ARP	: Address Resolution Protocol
CPU	: Central Processor Unit
DF	: Do Not Fragment
DNS	: Domain Name Server
DOS	: Denial Of Service
DDOS	: Distributed Denail Of Service
DHCP	: Dynamic Host Configuration Protocol
FIN	: Finish
FTP	: File Transfer Protocol
GNU	: General Public License
HIDS	: Host Intrusion Detection System
HTTP	: Hypertext Transfer Protocol
ID	: Intrusion Detection
IP	: Intrusion Prevention
IDS	: Intrusion Detection System
IPS	: Intrusion Prevention System
ISO	: International Organisation for Standardization
IPV4	: Internet Protocol Version 4
IPV6	: Internet Protocol Version 6
ICMP	: Internet Control Message Protocol
IGMP	: Internet Group Management Protocol
LAN	: Local Area Network
MAC	: Media Access Control

MITM	: Man In The Middle
NAT	: Network Address Translation
NIDS	: Network İntrusion Detection System
OS	: Operating System
OSI	: Open Systems Interconnection
PSH	: Push
RST	: Restart
RARP	: Reverse Address Resolution Protocol
SNA	: Systems Network Architecture
SMB	: Server Message Block
SSL	: Secure Socket Layer
SYN	: Synchronise
SLIP	: Serial Line Internet Protocol
SMTP	: Simple Mail Transfer Protocol
TCP	: Transmission Control Protocol
TC97	: Transmission Control97
TOS	: Type Of Service
TLS	: Transport Layer Security
TTL	: Time To Leave
UDP	: User Datagram Protocol
URA	: Uganda Revenue Authority
WAN	: Wide Area Network
VPN	: Virtual Private Network

LIST OF FIGURES

LIST OF TABLES

ÖZET

Anahtar kelimeler: Ağ güvenliği, Ağ saldırıları, Akıllı güvenlik Duvarı ajan, Saldırı Tespit Sistemi, hizmet Saldırı, Iptablo.

Günümüzde elektronik banka, elektronik ticaret ve elektronik vergi uygulamaları gibi çok sayıda işlem internet üzerinden gerçekleştirilmektedir. Bu işlemler çeşitli riskler içermekte, kişi ve kurumları çeşitli bilgi sızmalarıyla mesul bırakarak hedef haline getirebilmektedir. Günümüzdeki en yaygın saldırılar "DOS" ve "Spoofing" saldırılarıdır. Bu konuda çok sayıda açık kaynak uygulama olması, saldırganların bu uygulamalarla firmaların kaynaklarına kolayca erişebilmesini sağlamıştır. Çoğu firma klasik güvenlik sistemlerinin bir parçası olan saldırı tespit sistemleri ve güvenlik duvarı kullanmaktadır. Bu sistemlerin kullanılmasına rağmen, klasik sistemlerin işlevsel eksiklikleri vardır. Örneğin güvenlik duvarları zararlı paketlerle normal paketleri birbirinden ayıramazlar. Saldırı tespit sistemleri atakları tespit edebilir, fakat yanlış alarm da verebilmektedir. Bu durum, "DOS" ve "Spoofing" saldırılarına karşı daha etkili bir sistem geliştirme ihtiyacını ortaya çıkarmıştır. Çalışmada güvenlik duvarları ile saldırı tespit sistemlerini bütünleştirilecek zeki bir etmen sistemi ele alınmıştır.

SUMMARY

Keywords: Network security, network attacks, an intelligent firewall agent, intrusion detection system, denial of service attacks.

A number of transactions like e-banking, e-commerce and e-taxations are carried out over the internet today. Some of these transactions pose security risks and have made various people and organizations become targets of attacks there by exposing them to lots of business liabilities such as data leakages and compliance. Today the most common forms of attacks are DOS and Spoofing attacks and this is mainly due to the availability of a number of open source software which can be used by attacker's to easily gain unauthorized access to company resources and as a result numerous systems have been victims of DOS and spoofing attacks. Most organizations have been deploying traditional network security mechanisms such as firewalls and IDSs to secure their systems. Despite deploying these security measures, networks are still prone to attacks since traditional network security mechanisms have shortcomings for example firewall systems do not have the ability to differentiate between legitimate and illegitimate packets sent to a network. IDSs can detect attacks but give out a lot of false alarms. This has therefore necessitated the need to come up with a much more efficient defense mechanism against these DOS and Spoofing attacks. The study proposed an intelligent firewall agent, and the intelligent firewall agent integrated a firewall and IDS systems for prevention and detection of attacks respectively. Also an expert system was integrated in the IDS so that to record the time an attack happened in seconds by so doing false alerts can be reduced and prevent network attacks.

CHAPTER 1. INTRODUCTION

1.1. Background of the Study

Internet is changing our way of communication, business mode, and even everyday life. Almost all the traditional services such as banking, power, medicine, education and defense are extended to Internet now. With this, the use of internet is growing at an exponential rate in the last decades and continues to develop in terms of dimension and complexity[1]. The United Nations released report whıch states that nearly 3 billion people had access to the Internet by the end of 2014 [2], [3] reports that globally 3.2 billion people are using the Internet by end 2015. With increased reliance on internet, complexity of network attacks has also increased significantly [4].

Nevertheless, the network security threats increase with internet evolution. For instance a mere 171 vulnerabilities were reported in 1995 which increased to 7,236 in 2007, already the number for the same for merely the third quarter of 2008 has gone up to 6,058. Apart from these a large number of vulnerabilities go unreported every year [1]. In most of these cases it takes time to discover a real crime maker and until this moment some of these attacks are still registered as unknown for example state that big companies like Sony group and Google were penetrated by sophisticated attacks by some computer hackers who called themselves "Anonymous" in 2011[5]. Another cyber-attack occurred in Uganda between 2010-2012 in Kampala District where hackers without authority accessed Uganda Revenue Authority (URA) computerized systems, databases and released goods that had not paid customs taxes causing financial loss amounting to Ugx2 billion to the government of Uganda [6].

Regionally cyber-attacks are estimated to have cost Asia pacific businesses $81bn in the past 12 months while firms in EU ($62bn) and North America ($61bn) are also counting the significant cost of attacks. Despite the clear risk only just over of half of firms surveyed said they currently have a cyber-security strategy in place [7].

With loads of personal, commercial, military, and government information transferred over the internet, organizations are deploying traditional security mechanisms like Firewalling, IDSs and IPSs placed at the Internet edge to guard against any attacks. These mechanisms are also used to protect the network from external attacks. Such mechanisms are no longer enough to secure the next generation Internet [8]. Security vulnerabilities are discovered every year with just about every firewall on the market. What is worse is that most firewalls are often misconfigured, unmaintained, and unmonitored turning them into electronic doorstops (holding the gates wide open) [9]. Organizations require a systematic approach for securing their networks and to address that the study proposed an intelligent firewall agent so as to detect and prevent attack(s) on networks. By using Iptables firewall packets sent into a network can be filtered to ascertain whether they are coming from legitimate or illegitimate sources.

1.2. Statement of the Problem

The sharp increase in use of internet in different sectors or organizations for their day to day transactions has led to increased number of network attacks. Network attacks are from organizational to individual levels and this is due to the availability of tools and software programs which can be used by hackers to penetrate networks. Due to this a number of organizations have been victims of both spoofing and DOS attacks. These network attacks affect organizations by subjecting them to numerous risks, financial loses and penalties given by responsible authorities which have affected their efficiency, performance and reputation since customers have lost confidence in them.

Spoofing and DOS attacks occur while traditional security tools such as firewalls and IDSs are in place. The relative stagnation of the existing traditional security technologies according to [10] and the incapability of adequately protecting networks against attacks has led to increased attacks and [11] agree that existing traditional systems are problematic as follows; Every year there various cases of cyber-attacks caused by firewalls because they cannot differentiate between genuine incoming and outgoing traffic [9]. Traditional firewalls have degenerated in terms of ability to resist an attack against them and protect hosts behind them [12].

In addition [13] explain that more challenges like the management of manually configured firewall rules are complex thus making firewalls error prone while [14] assert that traditional firewalls also rely on topology restrictions and controlled network entry points to enforce traffic filtering. In addition, [4] elaborate that firewall systems cannot differentiate between legitimate packets from attacker packets, also intrusion detection system has the packet differentiation ability though with a high false rate.

To address the flaws and draw backs in the existing traditional firewall systems as stated above, the study employed an intelligent firewall agent to add intelligence to the traditional firewall systems. On the other hand the study aimed at coming up with a more effective mechanism against DOS and ARP spoofing network attacks by integrating an IDS with iptables firewall and an expert system to detect, prevent attacks and learn the detection time of an attack on a network respectively therefore serving as a counter measure against denial of service and ARP spoofing attacks.

1.3. Objectives of the Study

The primary objective of the study was to design and implement an intelligent firewall agent as a defensive mechanism against denial of service and ARP spoofing attacks.

1.3.1. Secondary objectives of the study

a. The study was carried out to analyze denial of service attacks and ARP spoofing attacks.
b. The study was carried out to detect and prevent DOS and ARP spoofing attacks on a network.
c. The study was also carried out to add intelligence to the traditional firewall systems.

1.4. Justification of the Study

Recent incidents in cyberspace prove that network attacks can cause huge amounts of loss to governments, private enterprises, and the general public in terms of money, data confidentiality, and reputation. The research community has been paying attention to the network security problem for more than two decades [15] but no genuine and effective mechanism has been developed to defend organizations against network attacks. Most of the existing Intrusion detection systems implemented nowadays depend on rule-based expert systems where new attacks are not detectable [16]. An intelligent firewall agent if implemented would provide a better and efficient solution to increasing network attacks.

Network attacks have become intriguingly overpowering and unfortunately most organizations do not have an open idea or a perfect and precise framework for countering these threats. Despite the fact that attacks trend are getting complex and the existing attack pattern recognition, learning and mitigation techniques prevailing in today’s traditional network security systems are evidently getting outdated.

However, all these attacks occur while security tools such as firewalls and intrusion detection systems are in place. Firewalls in existing operational networks are often problematic [11], firewall systems cannot differentiate legitimate packets from attacker packets also IDS has the packet differentiation ability it has a higher false rate [4]. The potential damage to computer networks keeps increasing due to a growing reliance on the Internet and more extensive connectivity. Intrusion detection systems (IDSs) have become an essential component of computer security to detect network attacks that occur despite the best preventative measures being in place. A major challenge with current intrusion detection systems is that they give out numerous false positive and false negative alerts.

1.5. Significance of the Study

This study proposed implementation of an intelligent firewall agent as a defense mechanism against network attacks. The intelligent firewall agent proposed in this study successfully detected attacks on a network, used a preventative mechanism to prevent against DDOS and ARP spoofing network attacks using Iptables firewall to filter incoming and outgoing packets making use of the allow and deny rules approach. Furthermore, integrating an expert system during the detection stage and a decision making approach helped learn the detection time of an attack thus reducing on the false alerts, improved system performance and added intelligence to the system respectively.

This study proposed a four way approach of Detection, Prevention, End of Attack and Cancel Attacks which ensures effectiveness than other measures which were earlier implemented but do not seem to embrace all the four approaches of network defense mechanism at once. Therefore the intelligent firewall agent approach is of paramount importance in the development of security mechanisms to prevent network attacks and reduce false alerts.

CHAPTER 2. LITERATURE REVIEW

2.1. Overview of the Open System Interconnected Model (OSI)

In 1977 ISO established a subcommittee to develop an architecture for the definition, development, and validation of standards for distributed data processing systems, and to define the functionality needed for communication among application processes in heterogeneous computer systems [17]. In 1984 in order to aid network interconnection without necessarily requiring complete redesign, OSI reference model was approved as an international standard for communications architecture [18]. The OSI model is an abstract representation of the seven basic layers (as stated below and also shown in Figure 1.1, in top to bottom order) involved to solve the communication problem: Application, Presentation, Session, Transport, Network, Data-link and Physical layers [19].

Below are the functions of the different OSI layers; the application layer specifies how one particular application uses a network and contacts the application program running on a remote machine. The presentation layer deals with the translation and/or representation of data at the two end hosts of the communication. The OSI Session Layer Protocol provides session management, e.g. opening and closing of sessions. In case of a connection loss it tries to recover the connection. If a connection is not used for a longer period, the session layer may close it down and re-open it for next use. This happens transparently to the higher layers [19]. The Session layer provides synchronization points in the stream of exchanged packets. On the other hand, functions in the Session Layer are those necessary to bridge the gap between the services available from the Transport Layer and those offered to the Session users. The session Layer are concerned with dialogue management, data flow synchronization, and data flow resynchronization [20].

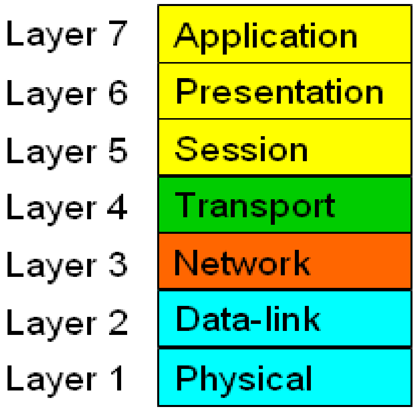


Figure 2.1. OSI Model [21].

2.2. The TCP/IP Protocol suite

The TCP/IP protocol suite is referred to as the Internet protocol suite and is the set of communications protocols that implements the protocol stack on which the Internet and most commercial networks run [22]. It is named after the two most important protocols in the suite: the TCP and IP protocols. The TCP/IP protocol suite like the OSI. [19], also states that the protocol suit, such as TCP/IP is the combination of different protocols at various layers. In addition, [19, 23] clarified that the TCP/IP is modelled in four (4) layers and this layered presentation leads to the term protocol stack, which refers to the stack of layers in the protocol suite. This layering is used for positioning (but not for functionally comparing) the TCP/IP protocol suite against others, such as Systems Network Architecture (SNA) and the Open System Interconnection (OSI) model.

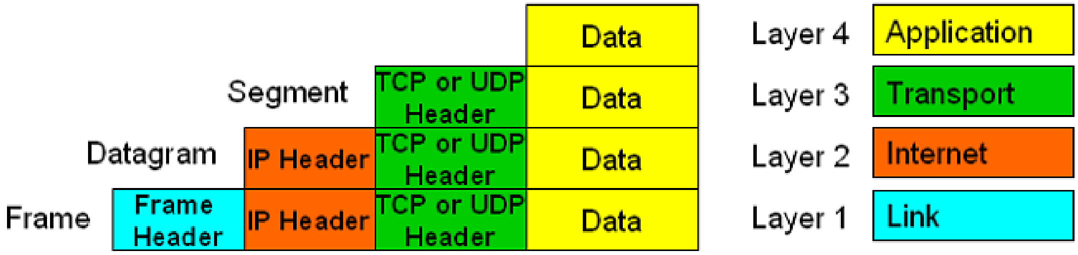


Figure 2.2. TCP/IP Protocol Stack and the Structure of a Data Packet[21].

About TCP/IP layering, [19] states that layering makes each layer responsible for a different facet of communication. [24] also explains that the basic idea of layering is that each layer adds value to services provided by the set of lower layers in such a way that the highest layer is offered the set of services needed to run distributed applications. Layering thus divides the total problem into smaller pieces.

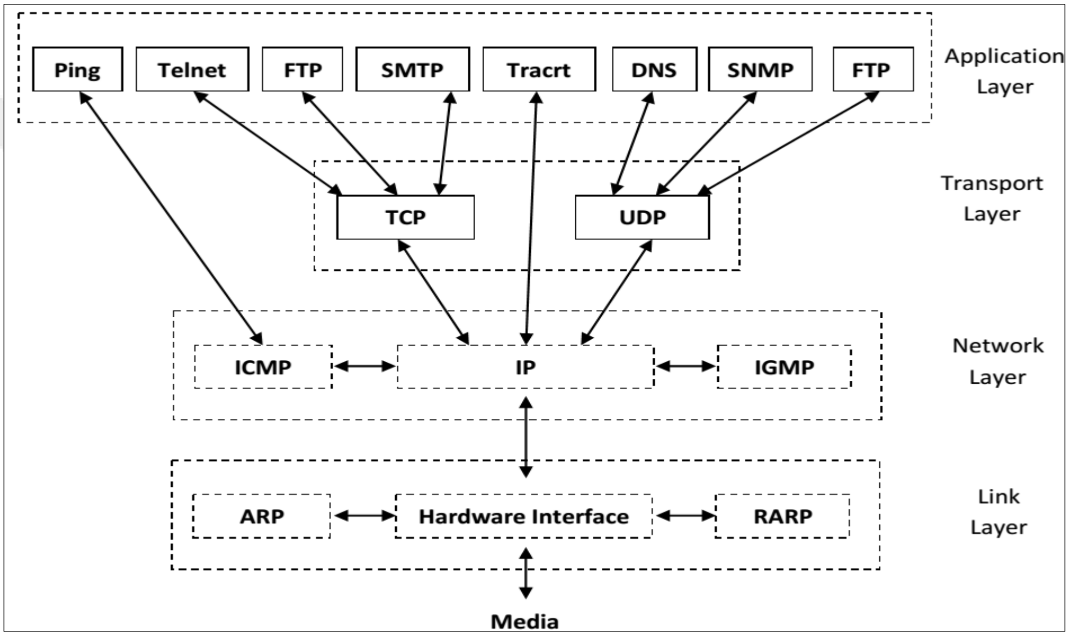


Figure 2.3. Shows different responsiblity of each layer [19].

2.2.1. Link layer

The link layer sometimes called the data link layer or network interface layer, normally includes the device drivers in the operating system and the corresponding network interface card in the computer. Together they handle all the hardware details of the physically interfacing with the cable (or whatever kind of media is being used). The important role of link layer concerns address resolution that provides mapping between two different forms of addresses with ARP and RARP protocols (see figure 2.4 for proper functionality; it has complete information of network interface cards, i.e. driver details and kernel information). It interprets between two systems in network for the sake of information of source address and destination address from software address to hardware address to send information on physical medium, because the kernel only recognizes the hardware address of network interface cards not the IP address or Physical address. Address resolution Protocols (ARP) translates an IP Address to a Hardware Address whereas Reverse Address Resolution Protocol (RARP) converts a hardware address to IP Address [19].

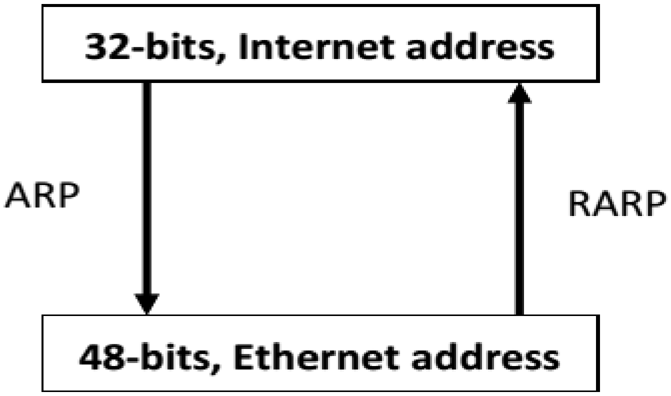


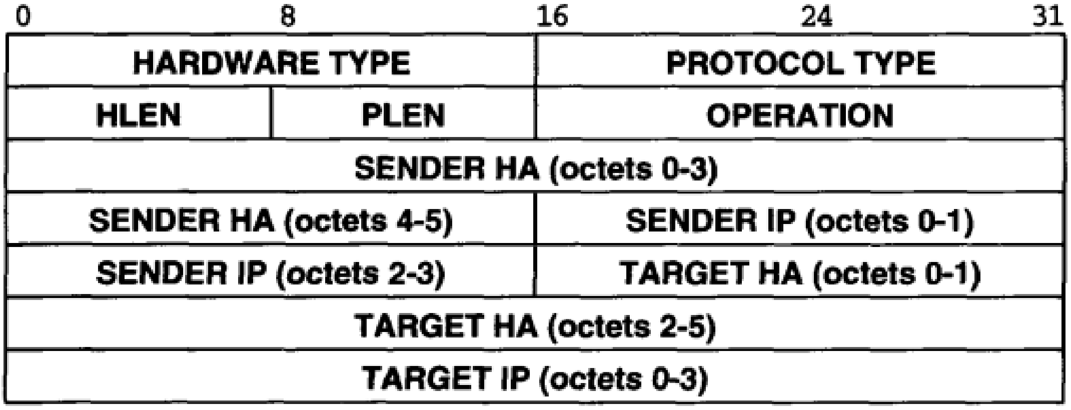
Figure 2.4. Resolution Protocols Working Scenarios [19].

2.2.1.1. Address resolution protocol

ARP provides a mapping between the two different forms of addresses [25]. In addition ARP as a protocol is used by the Internet Protocol (IP) [RFC826], specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol. And the protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer [26]. The Address Resolution Protocol (ARP) was developed to enable communications on an internetwork and Layer 3 devices need ARP to map IP network addresses to MAC hardware addresses so that IP packets can be sent across networks [27].

ARP operates in a way that; before a device sends a datagram to another device, it looks in its ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network. Each device compares the IP address to its own. Only the device with the matching IP address replies to the sending device with a packet containing the MAC address for the device (except in the case of "proxy ARP"). The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. The figure below illustrates the ARP broadcast and response process [28].

Furthermore [29] argues that unlike most protocols, the data in ARP packets does not have a fixed-format header. Instead, to make ARP useful for a variety of network technologies, the length of fields that contain addresses depend on the type of network. However, to make it possible to interpret an arbitrary ARP message, the header includes fixed fields near the beginning that specify the lengths of the addresses found in succeeding fields. In fact the ARP message format is general enough to allow it to be used with arbitrary physical addresses and arbitrary protocol addresses. The example in figure 2.5. Below shows the 28- octet ARP message format used on Ethernet hardware (where physical addresses are 48-bits or 6 octets long), when resolving IP protocol addresses (which are 4 octets long).



0	8	16	24 31
HARDWARE TYPE		PROTOCOL TYPE	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)	
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IP (octets 0-3)			

Figure 2.5. An example of the ARP/RARP message format [29].

2.2.1.1. Redirect address resolution protocol

RARP (defined in RFC903) is an early protocol for dynamic IP address assignment in Ethernet networks [29]. The TCP/IP protocol that allows a computer to obtain its IP address from a server is known as the Reverse Address Resolution Protocol (RARP) [30]. RARP often is used by diskless workstations because some network hosts, such as diskless workstations, do not know their own IP address when they are booted. To determine their own IP address, they use a mechanism similar to ARP, but now the hardware address of the host is the known parameter, and the IP address the queried parameter. RARP differs more fundamentally from ARP in a way that RARP server must exist on the network that maintains that a database of mappings from hardware address to protocol address must be pre-configured [27].

Also when it comes to the format of an RARP packet, [19] argues that RARP is almost identical to an ARP packet (Figure 2.5) and the only differences are that the frame type is 0x8035 for an RARP request or reply, and the op field has a value of 3 for an RARP request and 4 for an RARP reply. As with ARP, the RARP request is broadcast and the RARP reply is normally unicast.

2.2.2. Network layer

The network layer (sometimes called the internet layer) handles the movement of packets around the network. Routing of packets for example, takes place here. IP (Internet Protocol), ICMP (Internet Control Message Protocol) and IGMP (Internet Group Management Protocol) provide the network layer in the TCP/IP protocol suit [19].

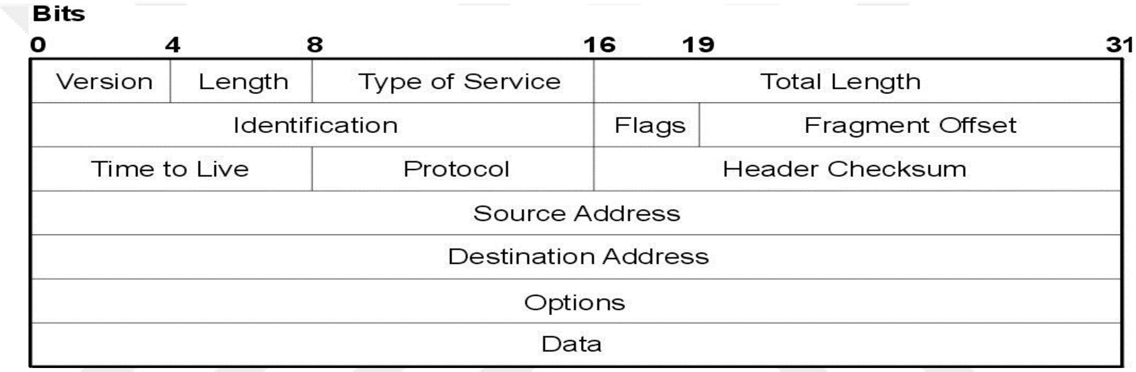
2.2.2.1. Internet protocol

The Internet Protocol (IP) is the standard network layer protocol of the Internet that provides an unreliable, connection-less, best-effort packet delivery service. The service is unreliable, because there are no guarantees that the IP datagram successfully gets to its destination [31]. The service is called unreliable because delivery is not guaranteed. The packet may be lost, duplicated, delayed, or delivered out of order, but the service will not detect such conditions, nor will it inform the sender or receiver. The service is called connectionless because each packet is treated independently from all others. A sequence of packets sent from one computer to another may travel over different paths, or some may be lost while others are delivered. Finally, the service is said to use best-effort delivery because the internet software makes an earnest attempt to deliver packets. That is, the internet does not discard packets capriciously; unreliability arises only when resources are exhausted or underlying networks fail [19, 29].

In figure 2.6 shows an IP format. The IP header format the most important bit is numbered 0 at the left and the least significant bit of a 32 bit value is numbered 31 on the right. The 4 bytes in the 32 bit value are transmitted in the order: bits 0-7, first, then bits 8-15, then 16-23 and bits 24-31 last. The current protocol version is 4, so IP is sometimes called IPv4. The header length is the number 32 bits words in the header, including any options and since this is a 4 bit field it limits the header to 60 bytes. The type of service field (TOS) is composed of a 3-bit precedence field (which is ignored today), 4 TOS bits, and unused bit that must be 0.The TOS bits are minimize delay, maximize throughput, maximize reliability, and minimize monetary cost. The total length field is the total length of the IP datagram in bytes [19].

Using total length field and the header length field, it becomes easy to know where the data portion of IP datagram starts and its length since this is a 16-bit field, the maximum size of an IP datagram is 65535 bytes. The identification field uniquely identifies each datagram sent by the host and it normally increments by one each time a datagram is sent.

The time- to- live field, or TTL sets an upper limit on the number of routers through which a datagram can pass. TTL limits the life time of the datagram and it also initialized by the sender to some value (often 32 or 64) and decremented by one by every router that handles the datagram. When this field reaches 0, the datagram is thrown away, and the sender is notified with an ICMP message. The checksum is calculated over the IP header only. It doesn’t cover any data that follows the header. ICMP, IGMP, UDP and TCP all have checksum in their own headers to cover their header and data [19].



Bits

0	4	8	16	19 31
Version	Length	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				
Data				

Figure 2.6. IP Datagram [19].

2.2.2.2. Internet control message protocol

ICMP was originally defined by RFC 792, but has since been updated by several other RFCs and is currently described by RFC 4884 [32] and ICMP is often considered part of the IP layer [19]. It is an invaluable tool when troubleshooting network problems and it communicates error messages and other conditions that require attention. ICMP messages are usually acted on by either the IP layer or the higher layer protocol (TCP or UDP). Some ICMP messages cause errors to be returned to user processes. ICMP messages are transmitted within IP datagrams [32].

Figure 2.7 below shows the format of an ICMP message. Although each ICMP message has its own format, they all begin with the same three fields: an 8-bit integer message **TYPE** field that identifies the message, an 8-bit **CODE** field that provides further information about the message type, and a 16-bit **CHECKSUM** field (ICMP uses the same additive checksum algorithm **as** IP, but the ICMP checksum only covers the ICMP message). In addition, ICMP messages that report errors always include the header and first 64 data bits of the datagram causing the problem [29].

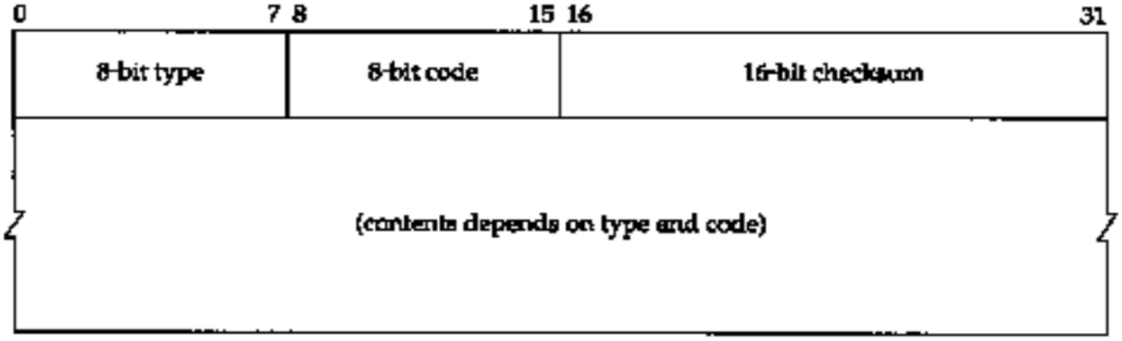
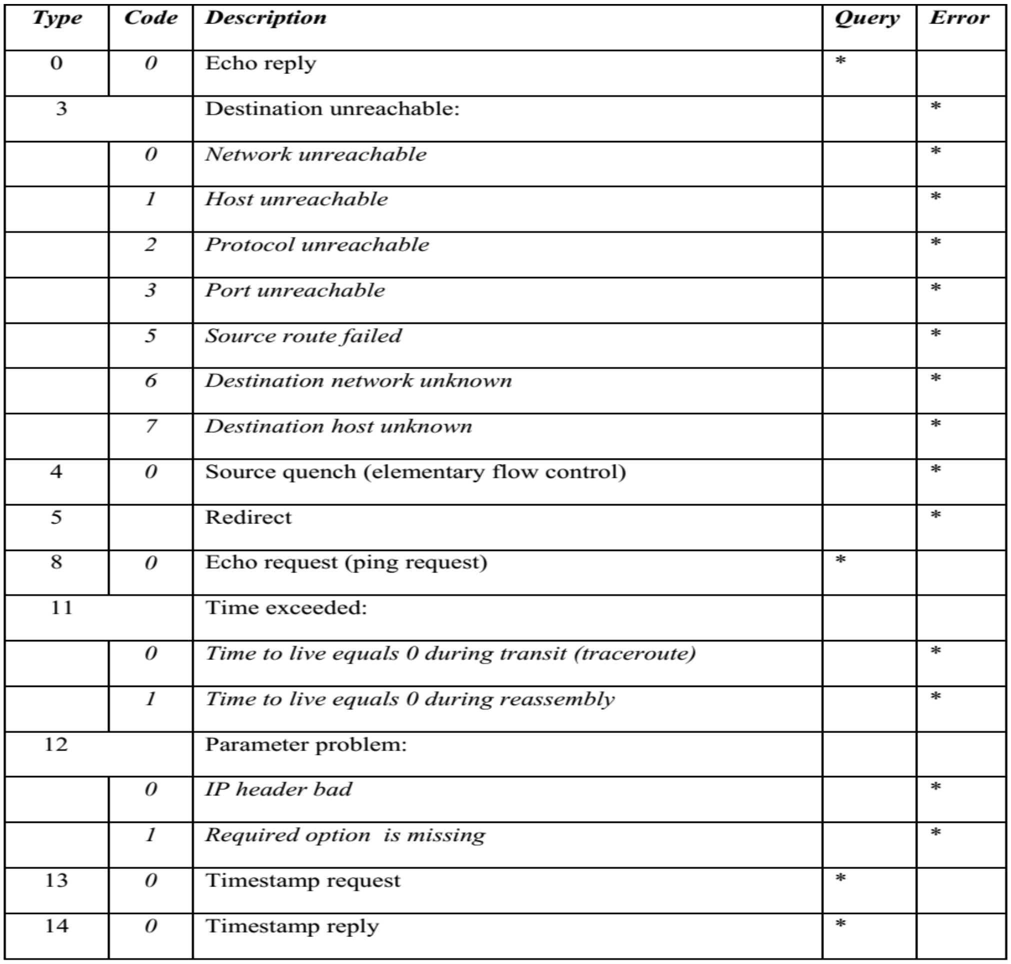


Figure 2.7. ICMP message [19].

The ICMP **TYPE** field defines the meaning of the message as well a**s** its format [29]. And there are 15 different values for the type field, which identify the particular ICMP message as shown in figure 2.8. Below;



Type	*Code*	*Description*	*Query*	*Error*
0	*0*	Echo reply	*	
3		Destination unreachable:		*
	0	*Network unreachable*		*
	1	*Host unreachable*		*
	2	*Protocol unreachable*		*
	3	*Port unreachable*		*
	5	*Source route failed*		*
	6	*Destination network unknown*		*
	7	*Destination host unknown*		*
4	*0*	Source quench (elementary flow control)		*
5		Redirect		*
8	*0*	Echo request (ping request)	*	
11		Time exceeded:		
	0	*Time to live equals 0 during transit (traceroute)*		*
	1	*Time to live equals 0 during reassembly*		*
12		Parameter problem:		
	0	*IP header bad*		*
	1	*Required option is missing*		*
13	*0*	Timestamp request	*	
14	*0*	Timestamp reply	*	

Figure 2.8. ICMP message types [33].

2.2.2.3. Internet group management protocol

IGMPv2 allows group membership termination to be quickly reported to the routing protocol, which is important for high-bandwidth multicast groups and/or subnets with highly volatile group membership [34]. Like ICMP, IGMP is considered part of the IP layer. Also like ICMP, IGMP messages are transmitted in IP datagrams. IGMP has a fixed size message data. IGMP messages are specified in the IP datagram with a protocol value of 2 [19].

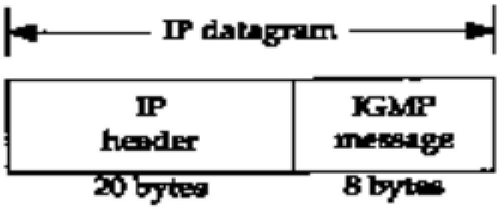


Figure 2.9. Encapsulation of an IGMP message within an IP datagram [19].

2.2.3. Transport layer

The transport layer provides a flow of data between two hosts, for the application layer above. In the TCP/IP protocol suite, there are two vastly different transport protocols: TCP (Transport Layer Protocol) and UDP (User Datagram Protocol). TCP provides a reliable flow of data between two hosts. Its concerned with things such as dividing the data passed to it from the application into appropriately sized chunks for the network layer below, acknowledging received packets, setting timeouts to make certain the other end acknowledges packets that are sent, and so on. Because this reliable flow of data is provided by the transport layer, the application layer can ignore all these details [29].

2.2.3.1. UDP

UDP is a transportation layer protocol, but it does not offer much more functionality other than IP. The checksum field in UDP header provides only a limited ability for error checking [35]. Figure 2.10 below shows a UDP header format.

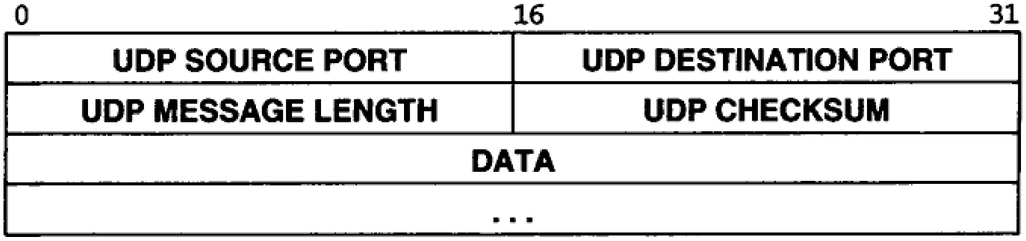


Figure 2.10. The format of fields in a UDP datagram [29].

In the TCP/IP protocol suite, the User Datagram Protocol or UDP provides the primary mechanism that application programs use to send datagrams to other application programs. UDP provides protocol ports used to distinguish among multiple programs executing on a single machine. That is in addition to the data sent, each UDP message contains a destination port number and a source port number, making it possible for the UDP software at the destination to deliver the message to the correct recipient and for the recipient to send a reply [29].

UDP doesn’t use acknowledgements to make sure messages arrive, it doesn’t order incoming messages, and it doesn’t provide feedback to control the rate at which information flows between the machines. Thus messages can be lost, duplicated, or arrive out of order. Furthermore, packets can arrive faster than the recipient can process them.

To summarize it all, the User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines. Thus an application program that uses UDP accepts full responsibility for handling the problem of reliability, including message loss, duplication, delay, out-of-order delivery, and loss of connectivity [29].

2.2.3.2. Transmission control protocol

To ensure reliable communications for applications and services that need them, TCP is available. It resides between IP and the application layer [31]. TCP provides a reliable, connection-oriented data stream delivery service [19] Connection oriented means the two applications using TCP (normally called the client and server) must establish a TCP connection with each other before they can exchange data.

Without options, TCP header occupies 20 bytes as shown in the figure 2.11. The source and destination port number is used to identify the sending and receiving processes. The sequence number is essential in keeping the sending and receiving datagram in proper order. There are six flag bits with the TCP header, namely URG, ACK, PSH, RST, SYN and FIN, each of them has a special use in the connection establishment, connection termination or other control purposes. Window size is advertised between communication peers to maintain the flow control [35].

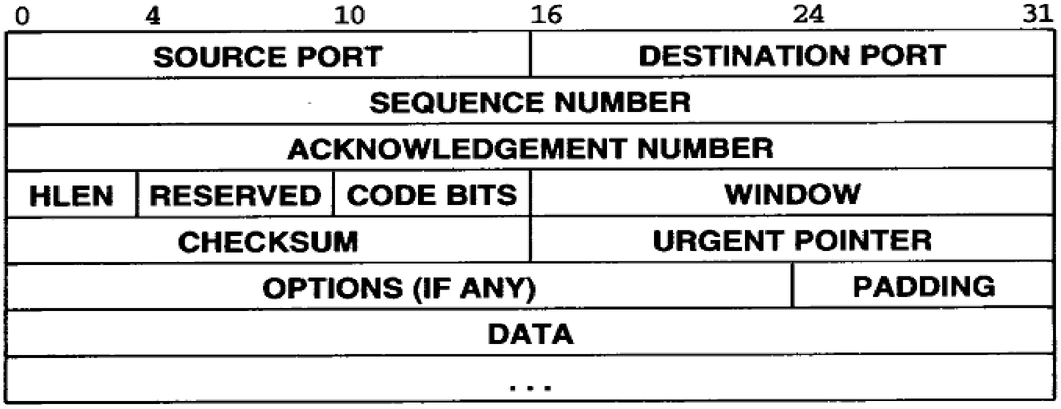


Figure 2.11. The format of a TCP segment with a TCP header followed by data [29].

2.2.4. Application layer

The Application Layer in TCP/IP groups the functions of OSI Application, Presentation Layer and Session Layer. Therefore any process above the transport layer is called an Application in the TCP/IP architecture. In TCP/IP socket and port are used to describe the path over which applications communicate. Most application level protocols are associated with one or more port number [20]. There are many common TCP/IP applications that almost all the implementations provides: Telnet for remote login, FTP, the File Transfer Protocol, and SMTP for electronic mail, SNMP and many other.

2.2.4.1 Simple mail transfer protocol

RFC 1425 defines the framework for adding extensions to SMTP [36]. The SMTP commands define the mail transfer or the mail system function requested by the users. The SMTP is used as the basis for most electronic mail (email). Email is the most popular Internet service, allowing people to communicate by exchanging electronic messages globally. These messages take anywhere from a few seconds to a couple of hours to be delivered. An added attraction is the relatively low cost of sending large messages. Combined, these benefits give users a convincing argument for access to email, and thus the connection of their systems to the Internet. For a full and easy to read description of SMTP the reader is urged to consult. It must be noted that SMTP is a developing protocol, and as such, new threats could evolve [37].

2.2.4.2. File transfer protocol

The FTP, RFC 959 enables the transfer of character and binary files across a network. The design philosophy does not dictate a specific host, operating system or file structure it is completely independent. An FTP server uses two TCP ports to transfer a file [36]. Control Connection is established on Port 21, and Data Connection on Port 20. The FTP client is free to choose any available port. FTP has become the standard for publishing software, data, and documents on the Internet. However Adobe Acrobat and Hyper Text Transfer Protocol (HTTP) using the Hyper Text Markup Language (HTML) are becoming popular for documents.

2.3. History of network security

Recent interest in security was fueled by the crime committed by Kevin Mitnick. Kevin Mitnick committed the largest computer-related crime in U.S. history. The losses were eighty million dollars ($8 million) in U.S. intellectual property and source code from a variety of companies. Since then, information security came into the spotlight. Public networks are being relied upon to deliver financial and personal information due to the evolution of information that is made available through the internet, information security is also required to evolve [38].

Due to Kevin Mitnick’s offense, companies are emphasizing security for the intellectual property. Internet has been a driving force for data security improvement. Internet protocols in the past were not developed to secure themselves. Within the TCP/IP communication stack, security protocols are not implemented. This leaves the internet open to attacks. Modern developments in the internet architecture have made communication more secure [38].

2.4. Network Security

“Network Security refers to all hardware and software functions, characteristics, features, operational procedures, accountability, measures, access control, administrative and management policy required to provide an acceptable level of protection for Hardware and Software, and information in a network” [39].

In addition, [40] explains that network security involves all activities that individuals, organizations, enterprises and institutions are undertaking to protect their value and on-going employment of assets and the integrity and continuity of operations. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions.

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources [41]. Network security is concerned with the concept of designing a secure network [42]. emphasize that network security is a main issue of computing because many types of attacks are increasing day by day [43]. As stated by [44], today, the Internet is an essential part of peoples‘ everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. According to recent sources the number of hosts connected to the internet has increased to almost 400 million and there are currently more than 1 billion users of the Internet.

Therefore, any disruption in the operation of the Internet can be very inconvenient for most of organizations' and people. Since almost all the traditional services such as banking, power, medicine, education and defense are extended to Internet now. The impact of Internet on society can be seen from the figure 2.12 below which shows exponential increase in number of hosts interconnected through Internet. Internet usage is growing at an exponential rate as organizations, governments and citizens continue to increase their reliance on this technology [1]. For instance internet is growing rapidly in the last decades and continues to develop in terms of dimension and complexity. At the end of 2014, 42.3% of the world population was connected to the network [8]

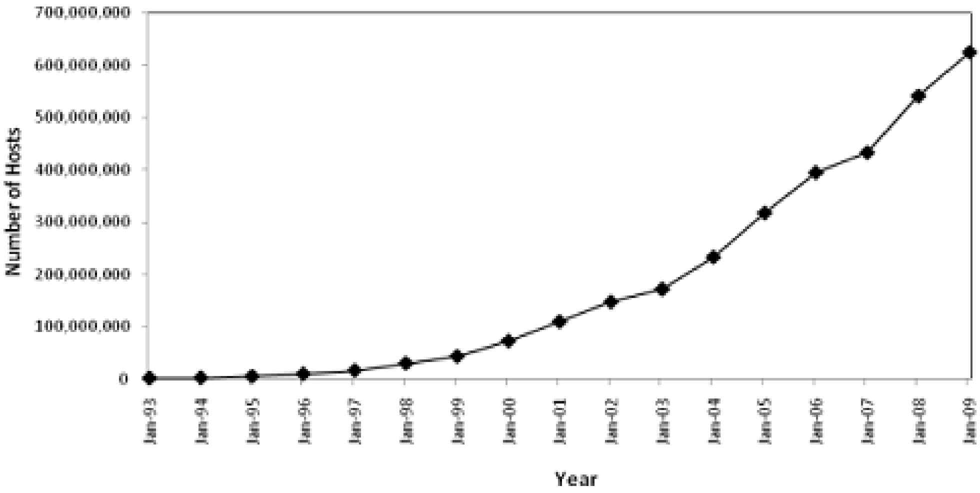


Figure 2.12. Internet Domain Survey Host Count [1].

Reports shows that the fastest growing cyber-threats involve attacks by nation states, competitors, and organized crime, though these remain much less common. According to White's findings, attacks by nation states were up 86% in 2014, with activity focusing mainly on the oil and gas, aerospace and defense, technology, and telecommunications sectors. Reports of security incidents attributed to competitors increased 64% compared with the previous year. Levels of theft by organized crime were particularly high in

Malaysia, India, and Brazil. Further she states that cyber-criminals also appear to be switching their focus to medium-size firms as large companies bolster their data security. Larger companies (those with gross annual revenues in excess of $1billion) said they have detected 44% more incidents than last year, while medium-size companies reported a 64% increase [45].

Recent incidents in cyberspace prove that network attacks have caused huge amounts of loss to governments, private enterprises, and the general public in terms of money, data confidentiality, and reputation. Security incidents cost businesses an average of $2.7 million each year, according to a survey by Price Water Coopers. The financial impact of breaches has also increased. The average reported loss from such incidents was up 34% in 2014 compared with the previous year. Furthermore, the number of organizations' reporting losses greater than $20 million nearly doubled. As many incidents go undetected or unreported, the true scale of the problem is even greater [45]. Despite the clear risks and loses incurred by various organizations, only just over of half of firms surveyed said they currently have a cyber-security strategy in place [45]. Also techniques which are used by organizations today to prevent attacks on the network are not enough and inefficient to fully solve the problem. Traditional security mechanisms like Firewalling, Intrusion Detection and Prevention Systems are deployed at the Internet edge are used to protect the network from external attacks. The available security tools are no longer enough to secure the next generation Internet [8]. Also that IDS have got a lot of false alarms also firewalls cannot differentiate between legitimate and illegitimate packets and there numerous security vulnerabilities discovered every year with just about firewall on the market. With all much importance attached to network security, a more systematic approach for securing the network is a must today [9].

Cyber-attacks are an increasingly significant danger for business. Not just cost in a financial sense but serious reputational damage can be inflicted if attacks undermine customer confidence. Despite this, nearly half of firms still lack a strategy to deal with the cyber threat and businesses cannot afford to be behind the curve on this threat since cyber-attacks can strike without warning and sometimes without the victim being immediately aware. The pressure from customers and clients cannot be ignored [7].

The number of potential security risks have increased at the same time that dependence on information technology has grown, making the need for a comprehensive security program even more important. Likewise, the job of those persons tasked with network security, often system administrators, has never been harder. The number of reported security incidents continues to grow and there is little indication that this trend will improve at any time in the near future [46]. In 2001, there were 52,658 reported incidents. By the end of the first quarter of 2002 there were already 26,829 incidents reported. A reported incident can be as simple as a single computer being compromised or as severe as a complete network compromise involving hundreds of client computers.

Unfortunately, many companies have stopped short of implementing a more secure “layered” approach to network security and have chosen to rely solely on the firewall/virus scanner approach. While firewalls and virus protection are necessary, by themselves they address only one portion of potential security risks and may contribute to a false sense of security. Although, network security is a critical requirement in emerging networks, there is a significant lack of security methods that can be easily implemented to ensure security of a network [40].

Till today when network security is mentioned, the general public is more often aware of security failures than of the technology available for secure communications. Viruses, worms, Trojan horses, denial-of-service attacks, and phishing are well known occurrences. Access controls, authentication, confidentiality, integrity, and non-repudiation, which are measures to safeguard security, are neither well known nor appreciated. However, when these security mechanisms are in place, users can have a degree of confidence that their communications will be sent and received as intended [47].

Therefore Network security has become a main issue of computing because many types of attacks are increasing day by day [43]. And this is because of the increased reliance on internet. In support for this [2] states that according to the United Nations released report nearly 3 billion people had access to the Internet by the end of 2014 furthermore, [3] reports that globally 3.2 billion people are using the Internet by end 2015.

Network Security is not only concerned with the security in the computers at each end of the communication chain but also with the security of a network as a whole. When transferring from one node to another node data the communication channel should not be vulnerable to attack. A hacker will target the communication channel, get the data, and decrypt it and reinsert a duplicate message. Though securing the network is just as important as securing the computers and encrypting the message [43]. And when developing a secure network, the following needs to be considered; [38, 43, 48].

2.4.1. Access

“This refers to the ability to control the level of access that individuals or entities have to a network or system and how much information they can receive” [49]. In addition [49] elaborates that access is the ability to limit and control the access to host systems and applications via communication links. For this, each entity trying to gain access must first be identified or authenticated, so that access rights can be tailored to the individuals.

A common threat that concerns many sites is unauthorized access to computing facilities. Access to network resources should only be permitted to authorized users. This is called authorized access. This access can take many forms, such as use of another user’s account to gain access to the network and its resources. In general, the use of any network resource without prior permission is considered to be unauthorized access. Therefore the network security policy should identify who is authorized to grant access to your services and determining what type of access these individuals can grant is important in identifying the cause of security holes as a result of users being granted excessive privileges. If you cannot control who is granted access to your system, it is difficult to control who is using your network. If you can identify the persons who are charged with granting access to the network, you can trace what type of access or control has been granted [50].

2.4.2. Confidentiality

“Confidentiality is the assurance that information is not made available or disclosed to unauthorized individuals, entities, or processes“[47]. Furthermore [50] defines “Confidentiality as the act of keeping things hidden or secret. He further emphasizes that It is an important consideration for many types of sensitive data”. Also [29] argues that the system must prevent outsiders from making copies of data as it passes across a network or understanding the contents if copies are made. Also [41] states that the other aspect of confidentiality is the protection of traffic flow from analysis. For example**,** a credit card number has to be secured during online transaction.

2.4.3. Authentication

“Authentication can be defined as the process of proving a claimed identity to the satisfaction of some permission-granting authority” [50]. Authentication systems are a combination of hardware, software, and procedural mechanisms that enable a user to obtain access to computing resources. Authentication mechanisms range from smart cards to biometric devices such as fingerprint readers, voice print readers, and retina scan devices. Secure user authentication is obtained through the encrypted exchange of the user’s security credentials or challenges.

Security credentials are used in this context to mean something that the authentication server knows about a particular user or device, for example, the knowledge of a valid user name, password, token, PIN, challenge, or in the case of an authentication device, the device’s ID [47]. And most systems, the user has to specify a password to their user account before they are allowed to log in. The purpose of the password is to verify that the user is who they claim to be, in other words; the password acts as a mechanism that authenticates the user. However, passwords can be stolen and someone else can impersonate the user. Because adequate measures are not taken as often as they should be, stolen passwords are the cause of a large number of security breaches on the Internet. By using the one time password un authorized users cannot access resources [50].

The one-time password system is designed to counter these types of attack and force a user to use a different password each time he or she logs in. This is accomplished by providing the user with a password that is different for each login, whether the login attempt is successful or not. As a result, it is not possible for the passwords to be re-used in a replay attack. In the S/Key system, the user generates a secret password to which a one-way function is applied. The secret password does not leave the user terminal, but what is sent is the hashed password [47].

2.4.4. Integrity

Integrity involves the unauthorized modification of information [51]. This could mean modifying information while in transit or while being stored electronically or via some type of media. [47] argues that integrity assures that data is not accidentally or deliberately modified in transit by replacement, insertion, or deletion. And to protect the integrity of information, one must employ a validation technique. This technique can be in the form of checksum, an integrity check, or a digital signature [47]. The process of verifying that the information that was sent is complete and unchanged from the last time it was verified. Information integrity is important for military, government, and financial institutions. It may also be important that classified information be undisclosed, whether it is modified or not modified. Information that is maliciously modified can create misunderstandings, confusion, and conflict [50].

2.4.5. Availability

“Availability is allowing legitimate users access to confidential information after they have been properly authenticated” [51] . When availability is compromised, the access is denied for legitimate users because of malicious activity such as denial of service (DOS) attacks. And Availability ensures that resources or services must be available at all time when needed [52].

2.4.6. Non-repudiation

“Non-repudiation refers to protection against an individual denying sending or receiving a message” [47]. The non-repudiation service may take one or two forms:

a. Non-repudiation with proof of origin: The recipient of the data is provided with a proof of the origin of data. This proof will protect the recipient against any attempt by the sender to falsely deny sending the data or its original content. The sender cannot deny that he sent the message, nor can the sender deny its original content.

b. Non-repudiation with proof of delivery: The sender of data is provided with proof of delivery of data. This proof will protect the sender against any subsequent attempt by the recipient to falsely deny receiving the data or its original content.

Network security [43] starts with authorization, commonly with a username and a password. Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, modification in system, misuse, or denial of a computer network and network-accessible resources. Basically network security involves the authorization of access to data in a network, which is controlled by the network admin.

Furthermore, [43] advise that when considering about network security, it is emphasized that the complete network is secure. It does not only concern with the security in the computers at each end of the communication chain. When transferring from one node to another node data the communication channel should not be vulnerable to attack. A hacker will target the communication channel, get the data, and decrypt it and reinsert a duplicate message.

Though securing the network is just as important as securing the computers and encrypting the message. It is important to have a well-conceived and effective network security policy that can guard the investment and information resources of your company. A network security policy is worth implementing if the resources and information your organization has on its networks are worth protecting [43]. Since most organizations have sensitive information and competitive secrets on their networks, these should be protected against vandalism in the same manner as other valuable assets such as corporate property and office buildings [50].

The Model of network Security Situation; they explain that the network environment is very complex, it consists of all kinds of computers, operating systems, services and programs, while it also can be simply concluded as a system to transfer data. By defining its single working pattern as transferring data, the network environment can be simplified as a world made up of many castles which are connected with highways, in the castle there are some houses filled with gold coins and some workers working in the house, their work is very easy, just send letters out to another place that connected by roads. If the workers make mistake then they may send out a gold coin, if they have weakness then they may invite in some intruders who are intended to steal the gold coins or be controlled by someone to send out the gold coins. And in this model Zhang considers the local networks as a castle, views the computers as houses and views the operating system and programs as the workers. The network security problems are abstracted as a mission to keep the gold safe in its house [53].

Like the locks used to help keep tangible property secure, computers and data networks need provisions that help keep information secure. Security in an internet environment is both important and difficult. It is important because information has significant value information can be bought and sold directly or used indirectly to create new products and services that yield high profits. Security in an internet is difficult because security involves understanding when and how participating users, computers, services, and networks can trust one another as well as understanding the technical details of network hardware and protocols. Thus he argues that security is required on every computer and every protocol; a single weakness can compromise the security of an entire network [29].

An effective network security plan can be developed with the understanding of security issues, potential attackers, need level of security, and factors that make a network vulnerable to attack. To lessen the vulnerability of the computer on the network there are many products available. These tools are encryption, authentication mechanisms. IDS, security management and firewalls. Businesses throughout the world are using a combination of some of these tools. ”Intranets” are both connected to the internet and reasonably protected from it. The internet architecture itself leads to vulnerabilities in the network. Understanding the security issues of the internet greatly assists in developing new security technologies and approaches for networks with internet access and internet security itself. The types of attacks through the internet need to also be studied to be able to detect and guard against them [54].

Knowing the network is very crucial when keeping the network secure since it is not possible to protect anything unless one clearly understands WHAT one wants to protect. So organizations of any size should have a set of documented resources, assets and systems. Each of these elements should have a relative value assigned in some manner as to their importance to the organization. Examples of hardware that should be considered are servers, workstations, storage systems, routers, switches, hubs, network and Telco links, and any other network elements such as printers, UPS systems and HVAC systems. Other important aspects of this task included documenting equipment location and any notes on dependencies. For instance most computers will rely on power backup systems such as UPSs which themselves may be part of the network if they are managed. Environmental equipment such as HVAC units and air purifiers may also be present [55].

In addition, [55] states that understanding different threats is the next step after knowing the network and this can be very helpful in identifying the potential “threats”. And threats can come from both internal and external sources. They may be human based, automated or even no intentional natural phenomenon. The latter might more appropriately be categorized under system health threats as opposed to security threats, but one issue can lead to the other. One example is a power outage to a burglar alarm. The power outage could be intentional or through some natural event such as a lightning strike. In either case security is diminished.

2.5. Network Attacks

“A Network attack is usually defined as an intrusion on your network infrastructure that will first analyze your environment and collect information in order to exploit the existing open ports or vulnerabilities” [56]. This may include as well unauthorized access to your resources, in such cases where the purpose of attack is only to learn and get some information from your system but the system resources are not altered or disabled in any way.

How serious a particular type of attack is depends on two things: how the attack is carried out, and what damage is done to the compromised system. An attacker being able to run code on his machine is probably the most serious kind of attack for a home user. For an e-commerce company, a DOS attack or information leakage may be of more immediate concern. Each vulnerability that can lead to compromise can be traced to a particular category or class of attack. The properties of each class gives a rough feeling for how serious an attack in that class is, as well as how hard it is to defend against. Attacks can lead to anything from leaving your systems without the ability to function, to giving a remote attacker complete control of your systems to do whatever he pleases.

A useful means of classifying security attacks is in terms of Active and Passive attacks. [39, 52, 57]. “A passive attack attempts to monitor the information from the system but does not affect system resources”. “An active attack attempts to harm system resources and their operations”.

2.5.1. Passive attacks

A passive attack monitors unencrypted traffic and looks for clear-text passwords and sensitive information that can be used in other types of attacks [39]. Though Passive attacks do not disrupt the normal operation of a network, it captures information about structure of a network and types of topology and the attacker snoops the data exchanged in a network without altering it. The requirement of confidentiality can be violated with passive attacks if an attacker is also able to interpret the data gathered through snooping [58]. Passive attacks include traffic analysis, monitoring of unprotected communications, decrypting weakly encrypted traffic, and capturing authentication information such as passwords [39]. In addition passive interception of network operations enables adversaries to see upcoming actions. Passive attacks result in the disclosure of information or data files to an attacker without the consent or knowledge of the user [39] on addition, [59], stated that with a passive attack a system is monitored and sometimes scanned for open ports and vulnerabilities. The purpose is solely to gain information about the target and no data is changed on the target.

2.5.1.1. Monitoring and eavesdropping

This is the most common attack to privacy. By snooping to the data, the adversary could easily discover the communication contents. When the traffic conveys the control information about the sensor network configuration which contains potentially more detailed information than accessible through the location server, the eavesdropping can act effectively against the privacy protection [60]. Interception of communications by an unauthorized party is called eavesdropping. Passive eavesdropping is when the person only secretly listens to the networked messages. On the other hand, active eavesdropping is when the intruder listens and inserts something into the communication stream. This can lead to the messages being distorted. Sensitive information can be stolen this way [61].

2.5.1.2. Traffic analysis

In this type of attack an attacker tries to sense the communication path between the sender and the receiver, this way the attacker finds the amount of data travelling between the route of the sender and the receiver and there is no alteration in data by the traffic analysis. Even when the messages transferred are encrypted, it still leaves a high possibility analysis of the communication patterns [62]. Sensor activities can potentially reveal enough information to enable an adversary to cause malicious harm to the sensor network [60].

2.5.2. Active attacks

"An active attack attempts to alter or destroy the data being exchanged in the network thereby disrupting the normal functioning of the network" [58]. The principle of active attacks is that in an active attack a malevolent third party manipulates a response within a legitimate session in a way that tricks the client into issuing an unwanted request (unknown to the user) that discloses sensitive information. The attacker can then apply a regular passive attack on this information. It is important to emphasize that this is made possible by a design flaw not an implementation error or bug [63].

In an active attack, the attacker tries to bypass or break into secured systems [39]. This can be done through stealth, viruses, worms, or Trojan horses. Active attacks include attempts to circumvent or break protection features, to introduce malicious code, and to steal or modify information. These attacks are mounted against a network backbone, exploit information in transit, electronically penetrate an enclave, or attack an authorized remote user during an attempt to connect to an enclave. Active attacks result into disclosure or dissemination of data files, DoS, or modification of data. Furthermore, [64] states that there are four types of active attacks that are available that is; replay, masquerade and modification of messages and denial of service etc.

In addition [52, 58] state that active attacks fall into two categories which is internal and external attacks.

1. Internal Attacks; an insider attack involves someone from the inside such as a disgruntled employee attacking the network. Insider attacks can be malicious or non-malicious. Malicious insider attacks intentionally eavesdrop, steal, or damage information; use information in a fraudulent manner; or deny access to other authorized users. No malicious attacks typically result from carelessness, lack of knowledge, or intentional circumvention of security for such reasons as performing a task [65]. Internal attacks act as valid nodes and perform malicious tasks. Internal attacks can access the communication link and advertise false routing and are very hard to detect as they appear like a normal node [52].

2. External Attacks; External attacks are not part of a network but are from some other entity or network. External attacks are often mere steppingstones leading to internal attacks when an outside attacker gains total control of a network node.

2.5.2.1. Denial-of-service attacks

“A denial of service (DOS) attack as an unambiguous effort by a horrible user to get through the capital of a server or a network thereby preventing rightful users from receiving services provided by the system” [66]. The sole purpose of DoS attacks is to disrupt the services offered by the victim. While the attack is in place and no action has been taken to fix the problem, the victim cannot be able to provide its services on the Internet. DoS attacks are really a form of vandalism against Internet services and take advantage of weaknesses in the IP protocol stack in order to disrupt Internet services [67].

In a practical networking environment DOS can be defined in different ways depending on the target (that is specific application or service) [68]:

a. Attacks against application servers: for example web servers, causing servers to be unavailable for public use.

b. Flooding network gateways and firewalls: for example flooding with thousands or millions of Transmission Control Protocol / Internet Protocol Suite (TCP/IP) packets causing either slowness in network activities or the network to become completely unusable till all packets are dropped from the network.

c. Attacking mail gateways: for example sending mail ware Simple Mail Transfer Protocol (SMTP) packets that can block email systems for a long period of time before they can be cleared.

On the other hand [69] emphasizes that a DoS attack takes place when availability to a resource is intentionally blocked or degraded by an attacker. Since DOS attacks aim at exhausting a resource in the target system which completely reduces or subverts the availability of the service provided [70]. In other words the DOS attack impedes the availability of the resource to its regular authorized users. The attack may concentrate on degrading processes, degrading storage capability, destroying files to render the resource unusable, or shutting down parts of the system or processes [69]. This can lead to the server being unable to service all the requests thereby denying offering service to legitimate requests [71].

“Normal” DoS attacks are being generated by a single host (or small number of hosts at the same location). The only real way for DoS attacks to impose a real threat is to exploit some software or design flaw. Such flaws can include, for example, wrong implementations of the IP stack which can crash the whole host when receiving a non-standard IP packet (for example ping-of-death). Such an attack would generally have lower volumes of data unless some exploits exist at the victim hosts which have not been fixed a DoS attack should not pose a real threat to high-end services on today's Internet [67].

On the other hand [58] argue that there exists many more ways to launch a DoS attack in a network which would not be possible in wired networks, DoS attacks can be launched against any layer in the network protocol stack. On the physical and MAC layers an adversary could employ jamming signals which disrupt the on-going transmissions on the wireless channel. On the network layer, an adversary could take part in the routing process and exploit the routing protocol to disrupt the normal functioning of the network for example an adversary node could participate in a session but simply drop a certain number of packets which may lead to degradation in the quality of service being offered by the network. On the higher layers an adversary could bring down critical services such as the key management service.

The key objective of a DDoS attack is to compile multiple systems across the Internet with infected zombies/agents and form botnets of networks. Such zombies are designed to attack a particular target or network with different types of packets. The infected systems are remotely controlled either by an attacker or by self-installed Trojans (e.g. roj/Flood-IM) that are programmed to launch packet floods [72]. Although not a requisite, DDoS attacks are usually aimed to exhaust network resources, which makes DDoS attacks often bandwidth consumption attacks. DDoS attacks are now performed by people with fine-tuned objectives in mind. The motives are numerous such as terrorism and the possible damages can be severe [73].

The concept of these attacks is straightforward: By flooding a given device whether it is an IDS, a Web server, or some other resource with enough traffic, an attacker can exceed its ability to process inbound requests in a timely fashion which causes the device to slow dramatically or in some cases become completely unavailable. Flooding DoS attacks are the most common type of DoS attacks because they require fewest resources to be conducted. The attacks seek to saturate a given system with hundreds, thousands, millions, or billions of connections, depending on the environment’s bandwidth or with an abnormally large amount of data flowing across a normal number of connections depending on the capacity of the attacked host, a flood DoS attack can be conducted by a single person using a single attacking system for example an attacker with a powerful server connected to an OC-3 line can easily take out a home-based system with more moderate hardware connected to the Internet via a DSL or a cable modem [74].

DDoS attacks do not rely on particular network protocol or system weakness. It simply exploits the huge resource asymmetry between the Internet and the victim. Since Internet architecture is open in nature, any machine attached to it is publically visible to another machines attached to enable the communication. The hacker or attacker community takes the unhealthy advantage of this open nature to discover any insecure machine connected to the Internet. The discovered machine is therefore infected with the attack code and the infected machine can further be used to discover and infect another machine connected and so on. The attacker thus gradually prepares an attack network called botnet. Depending upon the attacking code the compromised machines are called Masters/Handlers or zombies. Hackers send control instructions to masters which in turn control zombies [75]. The zombies under the control of masters/handlers transmit attack packets as shown in figure 2.13, which converge at victim to exhaust its resources [43].

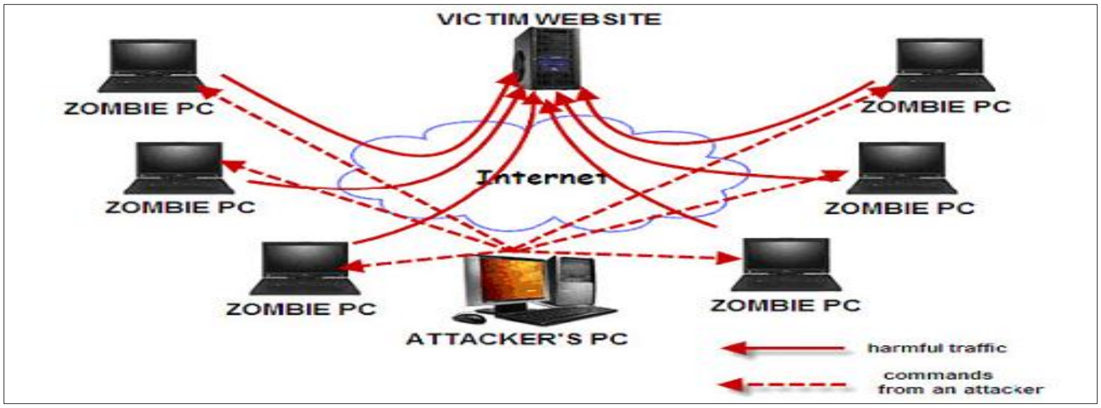


Figure 2.13. DDOS attack architecture [75].

Types of Denial of Service attacks:

To be able to understand DDoS attacks it is necessary to have a formal classification. DOS are classified in terms of the degree of automation, exploited vulnerability, attack rate dynamics and their impact [76].

a. SYN Flood Attack

The SYN flood is the most common type of flooding attack. It occurs when incoming connections repeatedly refuse to execute the third part of the TCP three-way handshake [77]. When a system (called the client) attempts to establish a TCP connection to a system providing a service (the server), the client and server exchange a sequence of messages. This connection technique applies to all TCP connections including telnet, Web, email, etc.

The process of how SYN flood attack occurs; the client system begins by sending a SYN message to the server, asking the server to open a connection. The server then acknowledges the SYN message by sending a SYN-ACK message to the client meaning it accepts to open the connection from the client (the ACK part) and asking if the client agrees to open the connection in the opposite sense (the SYN part). The client then finishes establishing the connection by responding with an ACK message to server. The connection between the client and the server is then open and the service-specific data can be exchanged between the client and the server. Figure 2.14 presents a view of this message flow [67]. Later system may exhaust memory, crash or be rendered otherwise inoperative. A relatively small flood of bogus packets will tie up memory, CPU, and applications resulting in shutting down a server.

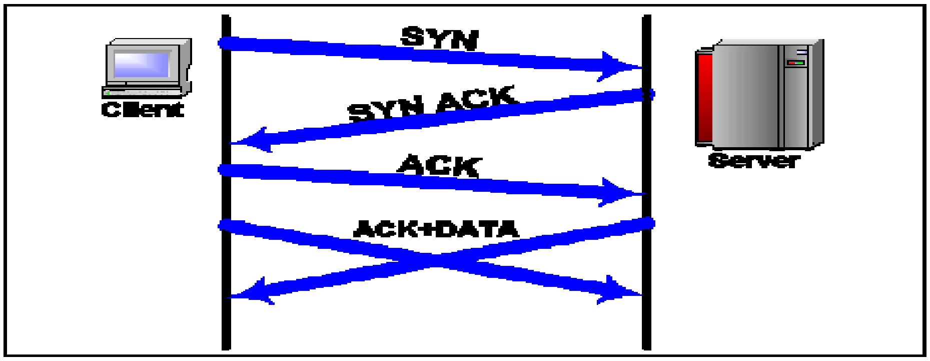


Figure 2.14. TCP three-way handshake [67].

b. UDP Flooding Attack

The UDP flood attack can be initiated by sending a large number of UDP packets to random ports on a remote host and this leads to depletion of available bandwidth for legitimate service requests to the victim system[70]. For a large number of UDP packets the victimized system will be forced into sending many ICMP packets eventually leading it to be unreachable by other clients. The attacker may also spoof the IP address of the UDP packets [78]. In addition a service on the host can crash by overwhelming it with information. The service cannot cope with all the available information and becomes unavailable for legitimate users [79]. This can sometimes cause systems connected to a network near a victim system to experience problems with their connectivity [80].

c. ICMP Flood

ICMP flood is a relatively simple attack where an attacker sends large number of ICMP Echo Request messages to the targeted machine. The aim of this attack is to overflow victim’s buffer or to take up the full bandwidth of a victim, so that no legitimate communication can be performed to and from the victim’s machine. Currently ping floods are not considered to be a very significant threat since many countermeasures have been developed against this particular type of attack. Counter measures against ping flood attacks include bandwidth limitation for ICMP connections, blocking ICMP messages at the edge router or turning off ICMP completely [81].

d. Ping of Death

The TCP/IP specification (the basis for many protocols used on the Internet) allows for a maximum packet size of up to 65536 octets (1 byte = 8 bits of data) containing a minimum of 20 bytes of IP header information and 0 or more bytes of optional information with the rest of the packet being data. It is known that some systems will react in an unpredictable fashion when receiving oversized IP packets. Reports indicate a range of reactions including crashing, freezing, and rebooting [67].

What makes the “Ping O’ Death” attack possible is the ability to send an echo request datagram with more than 65 507 octets of data and because of the way IP fragmentation is performed. IP fragmentation relies on an offset value in each fragment to determine the order in which the individual fragments should be reassembled. Therefore on the last fragment, it is possible to combine a valid offset with a suitable fragment size such that (offset+ size) > 65535. Since operating systems typically do not process the datagram until they have reassembled all the fragments, there exists the possibility of overflowing internal variables, and buffers which can lead to system crashes, reboots, kernel dumps, etc.

e. Echo/Chargen

CHARGEN is a simple service provided by most TCP/IP implementation under UNIX. It runs on both UDP and TCP port 19. For every incoming UDP packet the server just sends back a packet with 0 to 512 randomly selected characters. Another well-known service is ECHO which is running on UDP and TCP port 7. For each packet coming in the server just responses with whatever it has received. These two services are generally used for diagnosing purpose. However, they could be employed by a malicious denial-of-service type attack. Assuming a “chain” has been established between a CHARGEN service and an ECHO service, what will happen? Each of them will produce output continuously, leading to a very large number of packets among the network and thus a denial of service on the machines where the services are offered [35].

f. Smurf Attack

Smurf attack is a type of ICMP flood attack; this type of attack floods the target machine with the spoofed broadcast ping messages. An attacker sends a large quantity of the ICMP echo request packets to many different network broadcast addresses; all packets have a spoofed IP address of the target victim. Routers will forward the ICMP packet to all hosts, the hosts will try to reply to the ICMP request by sending a reply message to the victim. If there are many hosts in used networks, a victim will be effectively spoofed by a large amount of traffic [82]. This technique causes every computer to respond to the bogus ping packets and reply to the targeted computer which floods it. This technique is called a Smurf attack because the DoS tool that is used to perform the attack is called Smurf [83].

g. Teardrop Attack

The Teardrop is an old attack that relies on poor TCP/IP implementation that is still around. When the stack tries to reassemble packets, it cannot do it, and if it does not know how to toss these trash packet fragments out, it can quickly fail. Most systems know how to deal with Teardrops now and a firewall can block Teardrop packets in return for a bit more latency on network connections since this makes it disregard all broken packets. Of course if you throw a ton of Teardrop busted packets at a system, it can still crash. Many other variants such as Targa, SynDrop, Boink, Nestea Bonk, TearDrop2 and New Tear are available to accomplish this kind of attack [67].

h. Land Attack

A land attack consists of a stream of TCP SYN packets that have the source IP address and TCP port number set to the same value as the destination address and port number (i.e., that of the attacked host). Some implementations of TCP/IP cannot handle this theoretically impossible condition causing the operating system to go into a loop as it tries to resolve repeated connections to itself. Service providers can block LAND attacks that originate behind aggregation points by installing filters on the ingress ports of their edge routers to check the source IP addresses of all incoming packets. If the address is within the range of advertised prefixes, the packet is forwarded, otherwise it is dropped *[67]*.

2.5.2.2. Access attacks

Access attacks exploit known vulnerabilities in authentication services, FTP services, and web services to gain entry to web accounts, confidential databases and other sensitive information. Access attacks can consist of the following; Password attacks, Trust exploitation, Port redirection, Man-in-the-middle attacks, Social engineering, Phishing.

a. Man-in-the-Middle Attack

“A MITM attack is defined as an attack in which the intruder is able to read and write messages communicated between two parties of network without either party being conscious of this fact” [84]. With this type of attack there is an ability of the intruder to put himself between two communicating parties, a user (MS) and the network, enabling him for various actions including eavesdropping, modifying, deleting, reordering, replaying, and spoof signaling or user data. It becomes difficult for a user to understand whether they are connected to original secured connection or not. Since the certificate that is being passed during the connection setup is insecure, an attacker can easily modify the information in the certificate and leave the approval of the certificate to the user [85].

Furthermore [63] gives reasons why MITM attack is referred to as "active" rather than "passive", it is because of two essential differences in the nature of the attack:

- It is initiated by the attacker rather than the victim
- The target is entirely controlled by the attacker, rather than being limited by the extent of the victim's browsing activity.

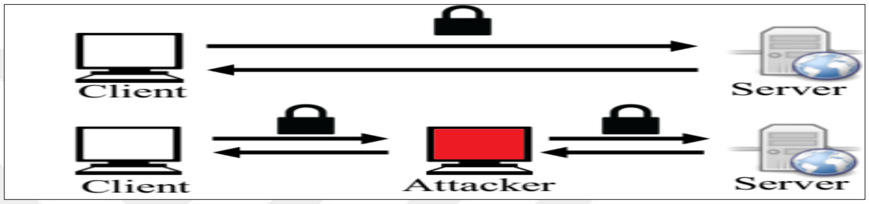


Figure 2.15. MITM Attack [85].

b. Password-Based Attacks

A common denominator of most operating system and network security plans is password-based access control. This means access rights to a computer and network resources are determined by who you are, that is user name and password. Older applications do not always protect identity information as it is passed through the network for validation. This might allow an eavesdropper to gain access to the network by posing as a valid user.
When an attacker finds a valid user account, the attacker has the same rights as the real user and therefore if the user has administrator-level rights, the attacker also can create accounts for subsequent access at a later time. After gaining access to a network with a valid account, an attacker can do any of the following: [86].

- Obtain lists of valid user and computer names and network information.
- Modify server and network configurations, including access controls and routing tables.
- Modify, reroute, or delete data.

c. Port redirection attacks

Once attackers have compromised a key target system such as firewall, they can use port redirection to forward al packets to a specified destination. The impact of this type of compromise is important to appreciate because it enables attackers to access to access any and all systems behind the firewall (or other target). Redirection works by listening on certain ports and forwarding the packets to a specified secondary target [9].

d. Trust exploitation attacks

Although it is more of a technique than a hack itself, trust exploitation as shown in figure 2.16 refers to an attack in which an individual takes advantage of a trust relationship within a network. The classic example is a perimeter network connection from a corporation. These network segments often house DNS, SMTP and HTTP servers. Because all these servers reside on the same segment, the compromise of one system can lead to the compromise of other systems because these systems usually trust other systems attached to the same network [9].

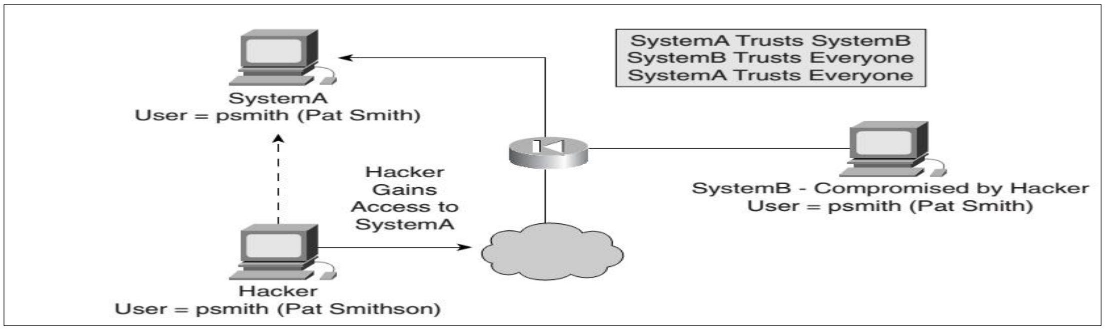

SystemA Trusts SystemB
SystemB Trusts Everyone
SystemA Trusts Everyone
SystemA
User = psmith (Pat Smith)
Hacker
Gains
Access to
SystemA
SystemB - Compromised by Hacker
User = psmith (Pat Smith)
Hacker
User = psmith (Pat Smithson)

Figure 2.16. Trust Exploitation.

2.5.2.3. Reconnaissance attacks

Reconnaissance attacks can consist of the following: Packet sniffers, Port scans, Ping sweeps and Internet information queries. A malicious intruder typically ping sweeps the target network to determine which IP addresses are alive. Later the intruder uses a port scanner to determine what network services or ports are active on the live IP addresses. From this information the intruder queries the ports to determine the application type and version, and the type and version of operating system running on the target host.

Based on this information, the intruder can determine whether a possible vulnerability exists that can be exploited. Using for example the Nslookup and Whois software utilities an attacker can easily determine the IP address space assigned to a given corporation or entity. The ping command tells the attacker what IP addresses are alive.

1. Port scanning

Port scanning provides list of open ports, closed ports and filtered ports. Through port scanning, attackers can find open ports and attack on services running on these ports. Network related details such as IP address, MAC address, router, gateway filtering, firewall rules, etc. can be known through this attack. Various port scanning techniques are TCP scanning, UDP scanning, SYN scanning, FIN scanning, ACK scanning, Window scanning etc. In Cloud scenario, attacker can attack offered services through port scanning (by discovering open ports upon which these services are provided) [87].

2. Packet Sniffers

A packet sniffer is a piece of hardware or software that monitors all network traffic [88]. Packet sniffers are used for collecting the data (request and response) which transverse the network among various network devices as well as end user [89]. It involves capturing packets from the network transmitted by other computers and reading the data content in search of sensitive information like passwords, session tokens and confidential information. This could be done using tools called network sniffers; these tools collect packets on the network and, depending on the quality of the tool, analyze the collected data like protocol decoders or stream reassembling [90]. The security threat presented by sniffers is their ability to capture all incoming and outgoing traffic including clear-text passwords and usernames or other sensitive material [91] and at the end it is these that the attacker can use to attack a system.

There are three types of sniffing methods. Some methods work in non-switched networks while others work in switched networks. The sniffing methods are: IP-based sniffing, MAC-based sniffing, and ARP-based sniffing [88, 92].

a. Identity spoofing (IP Address Spoofing) attacks

This is the original way of packet sniffing. It works by putting the network card into promiscuous mode and sniffing all packets matching the IP address filter. Normally the IP address filter isn’t set so it can capture all the packets. This method only works in non-switched networks [92]. When network card is set into promiscuous mode then host will be able to sniff all packets. A key point in the IP based sniffing is that it uses an IP based filter, and the packets matching the IP address filter is captured only [88].

b. MAC based sniffing

This is as like IP based sniffing. Same concept of IP based sniffing is also used here besides using an IP based filter. Here also a requirement of setting network card into promiscuous mode exists. Here in place of IP address filter a MAC address filter is used and sniffing all packets matching the MAC addresses [88].

c. ARP based sniffing

This method works a little different in that it does not put the network card into promiscuous mode [88]. Putting the network card in promiscuous mode is not necessary because ARP packets will be sent to the users. ARP based sniffing happens because the ARP protocol is stateless and therefore sniffing can be done on a switched network. To perform this kind of sniffing, the ARP cache1 of the two hosts that needs to be sniffed has to be poisoned first, having a user identified as the other host in the connection. Once the ARP caches are poisoned, the two hosts start their connection, but instead of sending the traffic directly to the other host it gets sent to us. We then log the traffic and forward it to the real intended host on the other side of the connection. This is called a man-in-the-middle attack [92].

3. Ping Sweep

A ping sweep (also known as an ICMP sweep) is a basic network scanning technique used to determine which of a range of IP addresses map to live hosts (computers). Whereas a single ping will tell you whether one specified host computer exists on the network, a ping sweep consists of ICMP ECHO requests sent to multiple hosts. If a given address is live, it will return an ICMP ECHO reply. Ping sweeps are among the older and slower methods used to scan a network. There are a number of tools that can be used to do a ping sweep, such as fping, gping, and nmap for UNIX systems, and the Pinger software from Rhino9 and Ping Sweep from Solar Winds for Windows systems. Both Pinger and Ping Sweep send multiple packets at the same time and allow the user to resolve host names and save output to a file [93].

2.6. Intrusion Detection Systems

“An IDS is a security system that monitors computer systems and network traffics and analyzes that traffic for possible hostile attacks originating from outside the organization and also for attacks originating from inside the organization” [16]. IDS are designed to detect any intrusion or hostile traffic in a network [94]. That is IDS are used to find out if someone has gotten into or is trying to get into your network and the most popular IDS is Snort [95] and IDSs can be software or hardware devices used to detect an attacks.

The main function of Intrusion prevention system is to identify malicious activity, log information about that activity, attempt to block/stop it and report it [42]. IDSs collect and inspect packets, looking for evidence of intrusive behaviors. As soon as an event is detected, an alarm is raised giving the security analyst an opportunity to react promptly and important to note, the main purpose of IDS is not to prevent attacks but to alert network administrator about the possible attacks so that they can be detected in time and hence their effect can be reduced. [96]. IDS products are used to monitor connection in determining whether attacks were being launched. Some IDS systems just monitor and alert of an attack, whereas others try to block the attack [39]. On the other hand, [42] argues that an IDS is also known as IPS. IPSs are considered the extension of IDSs because they both monitor traffic and network activities for malicious activities. The main Difference between IPS and IDS unlike IDSs, IPSs are placed at Network gateway level to prevent/block intrusions that are detected.

2.6.1. Approaches to intrusion detection system

There are two general approaches to ID namely [16, 97]; Misuse detection and anomaly detection. Misuse detection operate with prior prepared patterns, also called signatures, of known attacks that are used to detect intrusions by pattern matching on audit information. Anomaly detection deal with profiling user behavior. In other words, they define a certain model of a normal user activity. Any deviation from this model is regarded as anomalous.

2.6.2. Anomaly based approach

Anomaly detectors identify abnormal unusual behavior on a host or network [98]. They function on the assumption that attacks are different from legitimate activity and can therefore be detected by systems that identify these differences. Using statistical method for anomaly detection is one of the oldest techniques applied in IDS research. In this approach, the normal user behavior is first defined based on what is acceptable within the system usage policies.

Anomaly (or behavioral) detection is concerned with identifying events that appear to be anomalous with respect to normal system behavior [99]. A wide variety of techniques including data mining, statistical modeling and hidden Markov models have been explored as different ways to approach the anomaly detection problem. Anomaly based approach involves the collection of data relating to the behavior of legitimate users over a period of time and then apply statistical tests to the observed behavior, which determines whether that behavior is legitimate or not. Anomaly Based Detection has the advantage of detecting attacks which have not been found previously. The key element for using this approach efficiently is to generate rules in such a way that it can lower the false alarm rate for unknown as well as known attacks [97].

2.6.1.2. Signature based ıntrusion detection

Misuse (signature) based approach, is "detect what I know", when an attack has been discovered, the different steps of the attack are encoded in a signature, then the signature is stored in a database and finally it is used by the system to detect attacks [97]. Misuse detectors analyze system activity, looking for events or sets of events that match a predefined pattern of events that describe a known attack. Misuse detection technique is the most widespread approach used in the commercial world of IDSs. The basic idea is to use the knowledge of known attack patterns and apply this knowledge to identify attacks in various sources of data being monitored. Therefore, misuse detection based IDSs attempt to detect only known attacks based on predefined attack characteristics [98].

As a result [99] signature based systems are capable of attaining high levels of accuracy and minimal number of false positives in identifying intrusions. Little variation in known attacks may also affect the analysis if a detection system is not properly configured. Therefore signature based detection fails to detect unknown attacks or variation of known attacks and one of the motivating reasons to use signature based detection is ease in maintaining and updating preconfigured rules. These signatures are composed by several elements that identify the traffic for example in SNORT the parts of a signature are the header (e.g. source address, destination address, ports) and its options (e.g. payload, metadata), which are used to determine whether or not the network traffic corresponds to a known signature.

2.6.2. Classification of ıntrusion detection systems

It is classified into two basic type [16, 42]:- NIPS an HIDS.

2.6.2.1. Network based ıntrusion detection systems

“NIDS are intrusion detection systems that capture data packets traveling on the network media (cables, wireless) and match them to a database of signatures” [95]. NIPS monitors the entire network for suspicious activities like DoS attacks [99], port scans or even attempts to crack to into computers by analyzing protocol traffic like HTTP,TCP etc [42]. NIDS dedicate a specific network device to act as a network activity monitor and sensor, and when a malicious activity is detected it alerts the network and blocks the intruders from doing further damage to the network participating hosts.

The information collected from network is compared with known attacks for intrusion detection. NIDS has stronger detection mechanism to detect network intruders by comparing current behavior with already observed behavior in real time. NIDS mostly monitors IP and transport layer headers of individual packet and detects intrusion activity. NIDS uses signature based and anomaly based intrusion detection techniques. NIDS has very limited visibility inside the host machines. If the network traffic is encrypted there is no effective way for the NIDS to decrypt the traffic for analysis [99].

A NID [100] is not an access control mechanism (like a firewall). Nor is a firewall an ID device although firewalls can provide helpful insight on authorized access attempts. Just like an alarm system on a car, network based intrusion detection is also not going to stop an attack. It will alert that an attack is in process. There is a push in the industry to extend the functionality of IDS capabilities to interrupt communication sessions or modify access control lists to dynamically ‘defend’ against an attack. Interestingly enough, this capability actually creates a potential denial of service opportunity for a would-be hacker.

2.6.2.2. Host based ıntrusion detection systems

“HIDS refers to the class of IDS that reside on and monitor an individual host machine” [98]. HIDS monitors and analyzes the information collected from a specific host machine [99]. HIDS detects intrusion for the machine by collecting information such as file system used, network events, system calls, etc. HIDS observes modification in host kernel, host file system and behavior of the program. Upon detection of deviation from expected behavior, it reports the existence of attack. The efficiency of HIDS depends on chosen system characteristics to monitor.

HIDS detects intrusion for the machine by collecting information such as file system used, network events, system calls, etc. HIDS observes modification in host kernel, host file system and behavior of the program. Upon detection of deviation from expected behavior, it reports the existence of attack. The efficiency of HIDS depends on chosen system characteristics to monitor. Each HIDS detects intrusion for the machines in which it is placed [87]. These IDSs can look into system and application log files to detect any intruder activity. Some of these systems are reactive, meaning that they inform you only when something has happened. Some HIDS are proactive; they can sniff the network traffic coming to a particular host on which the HIDS is installed and alert you in real time [95].

2.7. Firewall systems

2.7.1. What is a firewall?

“A firewall is a networking system that helps in preventing unauthorized access of one’s computer over the internet (ie, It acts as a protection barrier between the system and the network)” [101]. While [102] elaborate that a firewall can be a hardware or software for defending the privacy, reliability, and accessibility of income and outcome packet over the network. In addition, [68] also states that different types of firewalls can be found on routers or even on dedicated network servers.

The firewall (sometimes referred to as a bastion host) is also a subsystem of computer software and hardware that intercepts data packets before allowing them into or out of a Local Area Network (LAN) [103]. A firewall makes decisions on whether or not data is allowed to pass based upon a security policy. For each packet of data, the firewall compares known components of the packet to a security rule set and decides if the packet should be allowed to pass. An efficient firewall is meant to intercept the data between the Internet and your computer. All data traffic must pass through it, and the firewall allows only authorized data to pass into the corporate network.

2.7.2. Firewalls in network security

The purpose of a firewall is to block traffic from the outside, but it could also be used to block traffic from the inside [54]. A firewall is also responsible for controlling access among devices, such as computers, networks, and servers [104]. Therefore the most common deployment is between a secure and an insecure network (for example, between the computers you control and the Internet), as shown in figure 2.17. A firewall may be placed to authorize all confident traffic and to deny all other requests, and in addition it may also be set to deny all messages of a particular kind except of specified network addresses or IP domiciles.

In addition, firewalls are intended to be a safeguard adjacent to effective endeavors of an assortment of security breaches [102]. Firewalls are a key part of the Internet infrastructure to protect users and network services against attackers. Basically, a firewall separates several network segments and it enforces a filtering policy. This filtering policy determines what packets are allowed to enter and to leave a given network segment. The filtering policy is expressed in an ACL that uses a vendor-specific low-level language.

The ACL contains a list of rules where each rule contains one or several packet selectors, for example the source and destination address, port source and destination, protocol type, and other fields available in the packet headers; and one action, for example accept, deny, log among many others [105].

Figure 2.17 shows a network consisting of an internal network (which is to be secured) and an external network (not trusted). The firewall controls access between these two networks, allowing and denying packets according to a security policy [104].

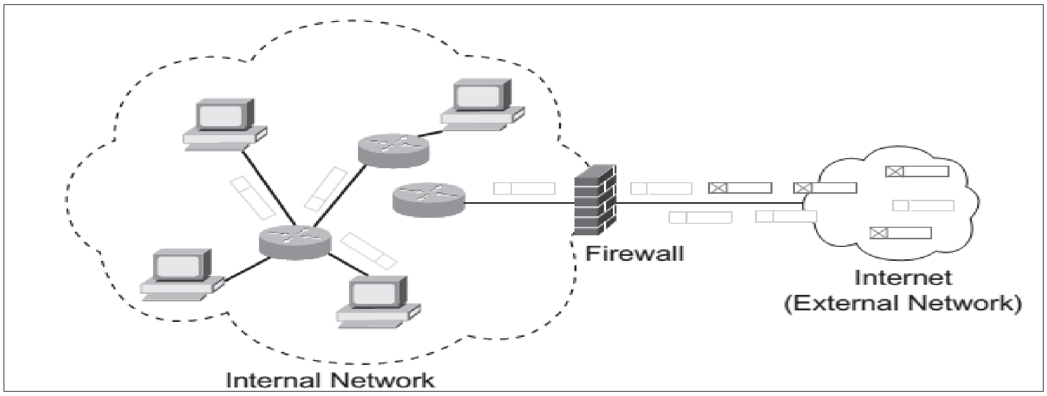


Figure 2.17. An example of a network consisting of an internal and an external network [104].

The common idea behind any firewall is to allow legitimate entities to access shared networked resources based on predefined policies. But the problem is that most firewalls do not know how to handle traffic unless it is predefined within the policy. In other words they lack the ability to learn from past experience and thus rely on human intervention. In most cases firewall administrators harden their firewalls by closing every port then setting rules to open certain ports as needed by the network users and applications. Hardening firewalls is effective in blocking illegal access to the network but cannot stop other sorts of network attacks by external and internal entities to the open ports. This is because they are considered legitimate activities by the firewall. In some cases even the most hardened firewalls can fall into an attacker's trap by responding to the attack packets rather than dropping them and continuing to process normal network activities [68].

Firewalls use security policies to inspect incoming and outgoing network traffic. A security policy consists of a set of filtering rules. Each filtering rule is defined by a set of filtering fields, and associated with an action to either block or forward a packet to its destination [106]. The last rule in a security policy is the default filtering rule which is usually assumed to be "Deny" [29, 106]. When a packet arrives at a firewall, a security policy is applied to determine the appropriate action. Actions include accepting the packet, which means the packet is allowed to travel to the intended destination. A packet can be denied, which means the packet is not permitted to travel to the intended destination (it is dropped or possibly is bounced back). The firewall may also log information about the packet, which is important to maintain certain services [104].

Firewall packet filtering [106] is performed in a sequential order starting from the first rule until a matching rule is found. If no matching rule is found, the packet is processed by the default rule. Thus, the computational complexity of the filtering process depends significantly on the length of each filtering rule as well as the depth of finding a matching filtering rule in the security policy. Hence, the order of the filtering rules, the order of the rule-fields, and the characteristics of the network traffic flow have all significant impact on the cumulative packet filtering time. And upon receiving a network packet, the firewall analyzes its characteristics (source address, destination address, port number, network status, actual data delivered, etc.) and determines whether to let it go through, drop it, delay it, or redirect it for further inspection.

Before building a firewall in preparation for connecting your network to the rest of the internet, it is important to understand exactly what network resources and services you want to protect. The Network Policy is a document that describes an organization’s network security concerns [50]. Furthermore [107] emphasized that when deploying firewalls in an organization, it is essential to verify that they are configured properly. Because firewall [108] is the foundation of enterprise network security, critical management task of firewall is required. Firewall configuration includes a set of large access control rules, each rules specify the source address, destination address, source port, destination port, one or more of the protocol ID and a proper function. This feature is a typical "accept" or "rejected".

Also [29] question was; How should a firewall be implemented? In theory, a firewall simply blocks all unauthorized communication between computers in the organization and computers outside the organization. In practice, the details depend on the network technology, the capacity of the connection, the traffic load, and the organization's policies. Thus, no single solution works for all organizations; building an effective, customized firewall can be difficult. To operate at network speeds, a firewall must have hardware and software optimized for the task. Fortunately, most commercial routers include a high-speed filtering mechanism that can be used to perform much of the necessary work.

A manager can configure the filter in a router to request that the router block specified datagrams. Alternatively, firewall implementation works well for an organization that has a single serial connection to the rest of the global Internet. Some sites have a different interconnection topology. For example, suppose a company has three or four large customers who each need to deposit or extract large volumes of information. The company wishes to have a single firewall, but allow connections to multiple sites [29].

Figure 2.18 below illustrates one possible firewall architecture that accommodates multiple external connections through a single firewall.

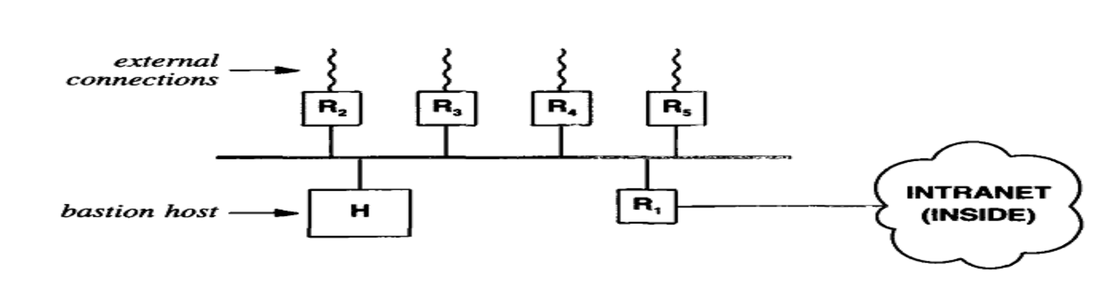


Figure 2.18. An alternative firewall architecture that permits multiple external [29].

2.7.3. Types of firewalls

Firewalls are classified into three main categories: [107, 109, 110]: packet filtering, circuit gateways, and application gateways. Commonly, more than one of these is used at the same time.

2.7.3.1. Packet filtering gateways

“A packet filtering gateway is defined as a firewall technique used to control network access by monitoring outgoing and incoming packet and allowing them to pass or half based on the source and destination internet protocol (IP) addresses, protocols and port” [111].

A packet filter [104] is the most basic type of a firewall since it only filters at the network and transport layers (layers two and three). Therefore a packet filter’s operations are similar to a network routers. The packet filter receives a packet, determines the appropriate action based on the policy, then performs the action on the packet. Packet filters only considers the IP addresses (layer two information), the port numbers (layer one information), and the transport protocol type (layer three information). Furthermore, since all this information resides in the packet header, there is no need to inspect the packet data (payload).

Packet filtering firewall only allows packets which are allowed as per your firewall policy. Each packet passing through is inspected and then the firewall decides to pass it or not [111]. Packet filtering gateways use the source or destination host to determine if the packet is allowed to pass the gateway [112]. And if the packet is allowed to or from the network it will be forwarded to the next hub as per the routing table information for this packet’s destination. On the other hand, if the packet is blocked it will be discarded.

The advantages of packet filters; they are fast, flexible, transparent (no changes are required at the client) and cheap. Most routers will provide packet filtering capabilities, and pure packet filter firewalls do not require powerful hardware [103]. This sort of “firewall” is normally quite cheap to implement, as today most routers have filtering capabilities. However, logging and alarming is normally not supported by routers and FTP, X11 and DNS services are not easy to implement properly. Another problem that could exist with using a packet filter as our firewall is the handling of IP fragments are handled. Normally, fragments are passed through the gateway as they are no threat to the inside system (either they can be reassembled and therefore the address/port of the first fragment was valid or they are dropped by the destination host) but if information leakage is a concern of yours, a packet filter may not be the best solution [68].

Furthermore, there are two types of packet filtering; Stateless packet filtering and Stateful packet filtering [113].

2.7.3.1.1. Stateless packet filtering

If the information about the passing packets is not remembered by the firewall, then this type of filtering is called stateless packet filtering [113]. In the stateless case, the filtering actions, such as accepting or rejecting packet flows, are taken according to a set of static configuration rules. These rules only pay attention to information contained in the packet itself, such as network addresses (source and destination), ports and protocol.

Indeed stateless packet filtering firewalls can block those packets that are not meeting the valid state machine of a given connection oriented protocol. As with stateless packet filtering, stateful filtering intercepts the packets at the network layer and verifies if they match previously defined security rules. Moreover, stateful firewalls keep track of each connection in an internal state table. Although the entries in this table varies according to the manufacturer of every product, they typically include source and destination IP addresses, port numbers and information about the connection status [113].

The main advantage of stateless firewalls is their filtering operations speed. However, since they do not keep track of state connection data, they fail at handling some vulnerabilities that benefit from the position of a packet within existing streams of traffic. Fortunately, Stateful firewalls solve this problem and improve packet filtering by keeping track of connection status [114]. In addition, these types of firewalls are not smart enough and can be fooled very easily by the hackers. And these are especially dangerous for UDP type of data packets. The reason is that, the allow/deny decisions are taken on packet by packet basis and these are not related to the previous allowed/denied packets [113].

2.7.3.1.2. Stateful Packet Filtering

These can be termed as smart firewalls. This type of filtering is also known as Dynamic packet filtering. Stateful packet filtering supports both connection and connectionless protocols (TCP, UDP, ICMP, and so on). Dynamic packet filtering monitors each connection and creates a temporary (time-limited) inbound filter exception for the connection. This allows blocking incoming traffic originating from a particular port number and address while still allowing return traffic from that same port number and address [113]. Stateful firewalls perform the same operations as packet filters but also maintain state about the packets that have arrived [104].

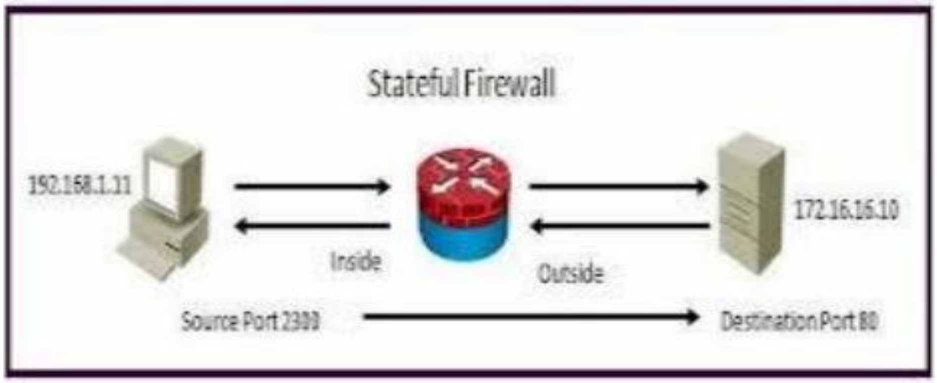


Figure 2.19. Stateful Packet Filtering [113].

Though stateful firewalls are problematic from the fault-tolerance perspective since they introduce a single point of failure in the network schema [105]. But they extend packet filtering capabilities by performing conformance checking upon the network traffic. Basically, stateful firewalls enforce that the communications between two peers evolve according to the protocol specification. In practice, stateful firewalls implement a finite state automaton for each supported protocol that determines what state-transitions are valid from the current state. Then, for each network packet, stateful firewalls check if it triggers a valid state-transition. This stateful capability allows the firewall administrator to modify the ACL to perform some action on the packets.

2.7.3.2. Circuilt proxies

In the circuit-level firewall, all connections are monitored and only those connections that are found to be valid are allowed to pass through the firewall. This generally means that a client behind the firewall can initiate any type of session, but clients outside the firewall cannot see or connect to a machine protected by the firewall. Stateful inspections usually occur at the Network Layer, thus making it fast and preventing suspect packets from travelling up the protocol stack [103].

The main difference between the circuit proxy and the packet filtering firewall is that the former is the addressee to which all communicators must address their packets. Assuming access has been granted, the circuit proxy replaces the original address (its own) with the address of the intended destination. It has the disadvantage of laying claim to the processing resources required to make changes to the header, and the advantage of concealing the IP address of the target system [109].

2.7.3.3. Application level proxies

An application gateway [113] goes one step beyond a packet filter. Instead of simply checking the IP parameters, it actually looks at the application layer data. Single application gateways are often called proxies, such as an SMTP proxy that understand the SMTP protocol. These check the data that is being sent and authenticate that the particular protocol is being used perfectly. Let’s say we were create an SMTP application gateway.

Further, [68] stated that application level firewalls allows network services (e.g. Telnet, FTP, etc.) to be established and used within predefined criteria controlled by firewall policies. However, even market leading firewalls such as Check Point Firewall-1 still lack a self-learning mechanism, in other words they cannot learn from past attacks dynamically. Firewalls are designed by humans and need human intervention in order to be kept up-to-date with latest security patches and rules configuration. The advantage with application gateways is that they are secure, but inefficient, either non-transparent to users and applications or are hard to be set up and manage. Only a limited set of applications is supported and special tailoring is needed for each one.

2.8. Firewall Technologies

Firewalls have additional capabilities one of them is Network Address Translation (NAT), Virtual Private Networks (VPN), Proxy [115].

2.8.1. Virtual private networking

“VPN is a connection that allows private data to be sent securely over a shared or public network such as the Internet” [115]. A VPN is constructed on top of existing network media and protocols by using additional protocols and usually encryption. VPNs are most often used to provide secure network communications across untrusted networks. For example, VPN technology is widely used to extend the protected network of a multi-site organization across the Internet, and sometimes to provide secure remote user access to internal organizational networks via the Internet. Two common choices for secure VPNs are IPsec6 and SSL/TLS

The two most common VPN architectures are gateway-to-gateway and host-to-gateway; Gateway-to-gateway architectures connect multiple fixed sites over public lines through the use of VPN gateways for example, to connect branch offices to an organization’s headquarters. A VPN gateway is usually part of another network device such as a firewall or router. When a VPN connection is established between the two gateways, users at branch locations are unaware of the connection and do not require any special settings on their computers [116].

Host-to-gateway, provides a secure connection to the network for individual users, usually called remote users, who are located outside of the organization (at home, in a hotel, etc.) Here, a client on the user machine negotiates the secure connection with the organization’s VPN gateway. For gateway-to-gateway and host-to-gateway VPNs, the VPN functionality is often part of the firewall itself. Placing it behind the firewall would require VPN traffic to be passed through the firewall while encrypted, preventing the firewall from inspecting the traffic. All remote access (host-to-gateway) VPNs allow the firewall administrator to decide which users have access to which network resources. This access control is normally available on a per-user and per-group basis; that is, the VPN policy can specify which users and groups are authorized to access which resources, should an organization need that level of granularity [116].

2.8.2. Network access control

NAT is the process of translating internal IP addresses to IP addresses that are visible to the external network. NAT is very often used with a special group of IP addresses (virtual IP, non-Internet-routable IP address), although it works with any IP address scheme. What NAT does is basically a one-to-one or a many-to-one IP address translation. An inside (local) IP address is mapped to an outside (global) IP address, meaning that an inside IP address is replaced by the appropriate outside IP address, and vice versa [115]. Network Address Translation includes the following steps:

- The IP address in the IP header is replaced with the new inside or outside IP address.
- The port numbers in the TCP/UDP header is replaced with the new port if port translation is enabled.
- The checksum for the IP packet is recalculated and checked for integrity.

The TCP header checksum must also be recalculated since this checksum is calculated using the new inside or outside IP address, new port (if applicable) and the payload (if applicable). Another common requirement for firewalls at the edge of a network is to perform client checks for incoming connections from remote users and allow or disallow access based on those checks. This checking, commonly called network access control (NAC) or network access protection (NAP), allows access based on the user’s credentials and the results of performing “health checks” on the user’s computer. Health checks typically consist of verifying that one or more of the following comply with organizational policy:

2.8.3. Proxy systems

Proxying provides Internet access to a single host, or a very small number of hosts, while appearing to provide access to all of your hosts. The hosts that have access act as proxies for the machines that do not doing what these machines want done. The proxy server evaluates requests from the client and decides which to pass on and which to disregard. If a request is approved, the proxy server talks to the real server on behalf of the client, and proceeds to relay requests from the client to the real server, and to relay the real server's answers back to the client. There are a number of advantages to using proxy services [115]:

2.8.4. Are the existing traditional network security mechanisms sufficient?

Nowadays some companies which recognize the importance of security adopt security systems such as firewalls, IDSs, and virus monitors. In this way, they integrate different security tools so as to secure their networks. [117] state that security mechanisms are used in different ways; virus monitors examine network traffic aiming to prevent malicious code from entering the network by detecting known malicious code patterns.

Virus monitors can only detect known viruses so new viruses can only be detectable after their pattern characteristics are being analyzed and are made available. IDSs are used to only detect and attacks. Furthermore, firewalls are used to guard and isolate connected segments of inter-networks. “Inside” network domains are protected against “outside” un-trusted networks, or parts of a network are protected against other parts.

Though it is not easy to use all security systems owing to high cost but it is not enough to protect a company’s system with only a single security system, because each security system has different features in present days, in which particularly DoS and virus attacks are becoming rampant. However, traditional security mechanisms have failed to fully deliver in as far as guarding networks against attacks is concerned since they have got a lot of challenges as stated below;

Virus monitoring approach is not effective since it can detect only known viruses. New viruses are only detectable after their pattern characteristics have been analyzed and are made available. And also for virus monitors and IDSs for them to be effective they need constant updating of new virus information [117].

[12] Argue that currently IDSs and firewalls have degenerated in terms of ability to resist attacks against them. [13] elaborate that traditional security mechanisms are also error prone. [14] state that traditional security mechanisms rely on topology restrictions and controlled network entry points to enforce traffic filtering.

Furthermore, [15] state that the current IDSs technology further proves sufficient for defending against casual attackers using well known techniques, but there is still a need to design tools to defend against sophisticated and well organized adversaries. In addition, traditional network security tools are not very sophisticated and rely on ad-hoc schemes and experimental work.

The current IDS technology may prove sufficient for defending against casual attackers using well known techniques, but there is still a need to design tools to defend against sophisticated and well organized adversaries. [9] emphasizes that IDSs have got a lot of false alarms also firewalls cannot differentiate between legitimate and illegitimate packets and there a lot of security vulnerabilities discovered every year with just about firewall on the market. With all much importance attached to network security, the study established that there was a need for a more systematic approach for securing networks.

As an effective defensive mechanism against network attacks and also to add intelligence to the traditional firewall, an intelligent firewall agent can protect a network against both DDOS and ARP spoofing attacks using a firewall to filter both incoming and outgoing packets making use of allow and deny rules approach. The intelligent firewall agent consists of a four way defense approach of detection, prevention, end of attack and cancel attacks. An expert system was also integrated in the detection stage. This could analyze the results gained after applying a preventative mechanism to learn the prevention time of attacks. And this increased the performance of the system.

Conclusively, with an intelligent firewall agent system provides effectiveness and efficiency than other measures which were earlier implemented but do not seem to embrace all the four approaches of network defense mechanism at once and keeping all this in mind, the study chose to implement a rules based intelligent firewall agent as the most suitable network defense mechanism.

CHAPTER 3. METHODOLOGY

In this section, an explanation of the proposed architecture is presented. The study proposed an intelligent firewall agent. The proposed system consisted of four components; First component was detection of attacks, Secondly applying prevention, thirdly end of attack and lastly cancel prevention as illustrated below.

3.1. Intelligent Firewall Agent Architecture

The architecture (figure 3.1 below) contained a firewall system as the preventative mechanism and an intrusion detection system for detecting attacks on a network. The IDS could monitor the network and incase of any attack it could trigger an alert to inform the system administrator that there is an attack on the network.

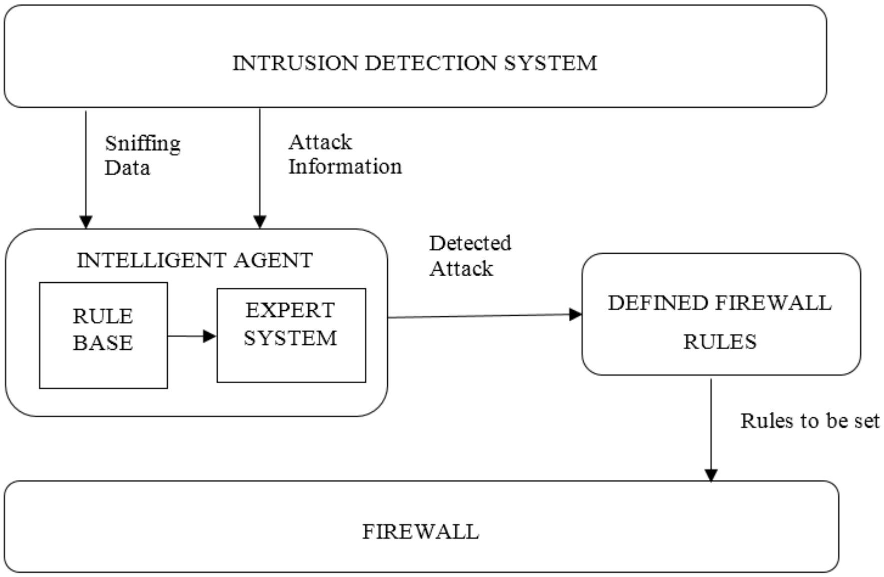


Figure 3.1. Design of Intelligent Firewall.

For detection, an IDS was applied to detect attacks on the network. The IDS was placed at the entry of the network (before) the firewall so that to monitor traffic flows from internal network to the external network or from external network to internal network. The IDS were used in a way that whenever any attack was launched on the network, the detection engine could send an alert to the network administrator. And this was inform of an alarm or alert message in order to inform that there is an intrusion or an attack. And the alert could provide all the relevant information concerning the attack type, time attack happened, IP address of the attacker.

On addition network sniffers were used to analyze and monitor packets on the network so that to find out any kind of anomalies in the transmitted data packets. With sniffing, researchers managed to legitimately capture data being transmitted. The sniff data output was analyzed in case of any anomalies in the captured packets, the attacker information could be analyzed later using rule sets.

With the help on an expert system, the time at which an attack occurred could be recorded to calculate the mean of packet sending for attacks. This increased performance in a way that I could record the rate of packets that can be sent during the process of detection and it also recorded the number of packets that can be sent in a given time. The time difference at which the packets were received and left the network was considered to find out suspicious packets. During Sniffing, the captured sniff output was fully analyzed so that the intelligent agent could find any anomaly in the packet headers in that obviously that could show that there is intrusion or any kind of attack on the network.

Once packets have been handled by all enabled preprocessors, they are handed off to the detection engine. The detection engine is the meat of the signature-based IDS in Snort. The detection engine takes the data that comes from the preprocessor and its plug-ins, and that data is checked through a set of rules [118]. If the rules match the data in the packet, they are sent to the alert processor. The signature based IDS function is accomplished by using various rule sets. Depending on what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. And incase an attack was found in the packets, an alert was triggered.

Since IDS were entirely used to detect any intrusion or attacks on the network, if at all there was an attack detected, according to the intelligent firewall system the next step we had to defend the network against the detected attacks. A preventative mechanism was applied so that to prevent attacks on the network. The study applied firewall system as the preventive mechanism to defend against attacks (DoS and ARP Spoofing).

Using firewall system, defined firewall rules were set. And these rule sets were formulated basing on Source IP address, Destination IP, Source Port, Destination Port, Packet size and Protocol. Here the aspect of intelligence in this system is reflected by the use of decision making system which can decide whether to block or allow traffic depending on the rules set.

In figure 3.2. Illustrates the detection and prevention system applied during this study.

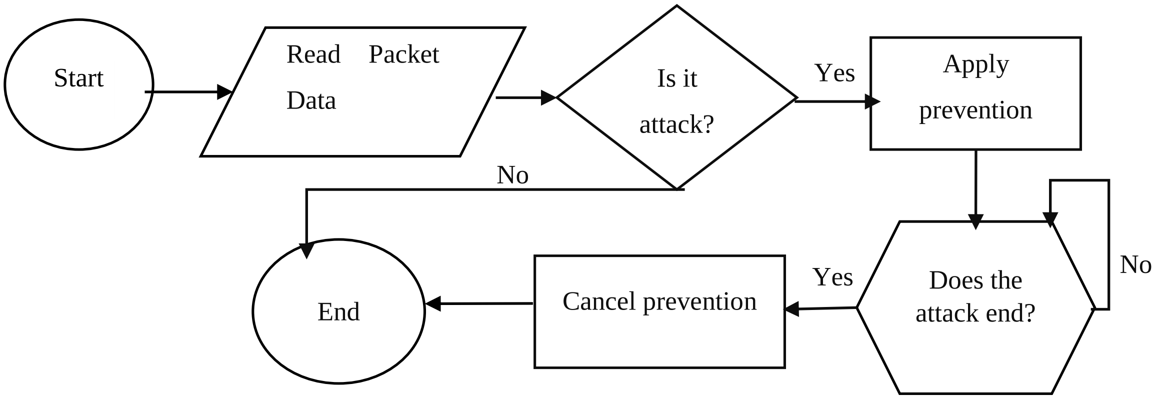


Figure 3.2. The Decision and Prevention Process

And this worked in a way that it could read the packets sent via the network to check whether in the transmitted packets there are suspicious ones. And if data packets were found suspicious, as explained before, that evidenced a malicious activity (attack). This called for a defense against the found attacks in the network. Thus a firewall rule based system is used. With this mechanism, incoming and outgoing packets could be fully analyzed using set rules. And only packets which match with the defined rule sets were allowed to go through the network while those packets which do not match rules are dropped. Hence by using this mechanism it successfully defended the network against attacks meaning it’s an intelligent firewall agent is an appropriate defensive mechanism against attacks.

Does the attack end? Yes, after applying the preventative mechanism (firewall rule sets) it could only allow genuine packets which matched the defined rule sets meaning with this the attack could end. And prior to the end of attack, the preventative mechanism is cancelled and finally the system ends and vice versa.

3.2. Proposed Tools

3.2.1. Snort

Snort is an open source network intrusion prevention and detection system (IDS/IPS) that combines the benefits of signature, protocol, and anomaly base inspection. Snort uses a set of rules to check for hostile packets in the network and then generate alerts to the network administrator [94]. Snort is a lightweight intrusion detection system which can easily be deployed on almost any node of a network with minimal disruption to operations. Lightweight IDSs are small, powerful, and flexible enough to be used as permanent elements of the network security infrastructure [119].

On addition, snort is a well-known and accepted IDS within network security communities and it was created by Martin Roesc in 1998 [94]. And this has left snort as the most popular NIDS and it is Open Source, which means that the original program source code is available to anyone at no charge and this has allowed many people to contribute to and analyze the programs construction. SNORT uses the most common open-source license known as the GNU General Public License [120]. The main aim of using snort and other IDSs is to effectively analyze all packets passing through the network without any packet drop [94].

3.2.1.1. Components of snort

Snort is logically divided into multiple components which work together to detect particular attacks and generate output in a required format from the detection system [95];

a. Packet Decoder
b. Preprocessors
c. Detection Engine
d. Logging and Alerting System
e. Output Modules

Figure 3.3 below shows the arrangement of snort components. Any data packet coming from the network enters the packet decoder. And on its way towards the output modules, it is either dropped, logged or an alert is generated.

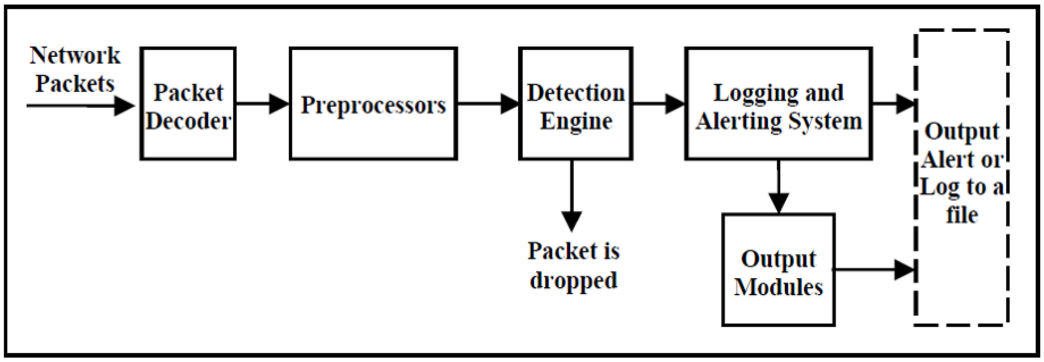


Figure 3.3.Components of Snort [121].

Packet decoder:

The packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed or to be sent to the detection engine. The interfaces may be Ethernet, SLIP, PPP and so on [95].

Preprocessor:

A preprocessor takes the raw packets and checks them against certain plug-ins (like an RPC plug-in, an HTTP plug-in, and a port scanner plug-in).These plug-ins check for a certain type of behavior from the packet. Once the packet is determined to have a particular type of “behavior,” it is then sent to the detection engine. Snort supports many kinds of preprocessors and their attendant plug-ins, covering many commonly used protocols as well as larger-view protocol issues such as IP fragmentation handling, port scanning and flow control, and deep inspection of richly featured protocols [118].

The detection engine:

The detection engine is the most important part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets.

If a packet matches any rule, appropriate action is taken; otherwise the packet is dropped. Appropriate actions may be logging the packet or generating alerts [95]. Depending upon how powerful your machine is and how many rules you have defined, it may take different amounts of time to respond to different packets. If traffic on your network is too high when Snort is working in NIDS mode, you may drop some packets and may not get a true real-time response.

The logging and alerting system:

After the Snort data goes through the detection engine, it needs to go out somewhere. If the data matches a rule in the detection engine, an alert is triggered. Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump- style files or some other form. Alerts can be sent to a log file, through a network connection, through UNIX sockets or Windows Popup (SMB), or SNMP traps. The alerts can also be stored in an SQL database such as MySQL and Postgres [118].

Alerts are any sort of user notification of an intruder activity. When an IDS detects an intruder, it has to inform security administrator about this using alerts. Alerts may be in the form of pop-up windows, logging to a console, sending e-mail and so on. Alerts are also stored in log files or databases where they can be viewed later on by security experts. And snort can generate alerts in many forms and are controlled by output plug-ins. Snort can also send the same alert to multiple destinations. For example, it is possible to log alerts into a database and generate SNMP traps simultaneously. Some plug-ins can also modify firewall configuration so that offending hosts are blocked at the firewall or router level [95].

The output:

Basically these modules control the type of output generated by the logging and alerting system. Depending on the configuration, output modules can send output messages to a number of other destinations. Commonly used output modules are:

- The database module is used to store Snort (Snort 2004) output data in databases, such as MySQL, MSSQL or Oracle,
- The SNMP module can be used to send Snort (Snort 2004) alerts in the form of traps to a management server,
- The Sending Server Message Block (SMB) alerts module can send alerts to Microsoft Windows machines in the form of pop-up SMB alert windows,
- The syslog module logs messages to the syslog utility (using this module you can log messages to a centralized logging server.)
- XML or CSV modules can be used to save data in XML or comma separated files. The CSV files can then be imported into databases or spreadsheet software for further processing or analysis.

3.2.1.2. Snort modes

Snort operates in two basic modes: packet sniffer mode and NIDS mode. It can be used as a packet sniffer, like tcp dump or snoop [95]. When sniffing packets, Snort can also log these packets to a log file. The file can be viewed later on using Snort or tcp dump. No intrusion detection activity is done by Snort in this mode of operation. Using Snort for this purpose is not very useful as there are many other tools available for packet logging.

When Snort is used in NIDS mode, it uses its rules to find out if there is any network intrusion detection activity. The rules contain the information that defines the who, where, and what of a packet, as well as what to do in the event that a packet with all the attributes indicated in the rule should show up. The first item in a rule is the rule action. The rule action tells Snort what to do when it finds a packet that matches the rule criteria. There are five available default actions in Snort: alert, log, pass, activate, and dynamic [115].

- Alert; generate an alert using the selected alert method, and then log the packet,
- Log; log the packet,
- Pass; ignore the packet,
- activate; alert and then turn on another dynamic rule,
- Dynamic; remain idle until activated by an activate rule, then act as a log rule.

Network intrusion detection mode:

In network intrusion detection mode, Snort does not log each captured packet as it does in the network sniffer mode. Instead, it applies rules on all captured packets. If a packet matches a rule, only then is it logged or an alert is generated. If a packet does not match any rule, the packet is dropped silently and no log entry is created. When you use Snort in intrusion detection mode, typically you provide a configuration file on the command line. This configuration file contains Snort rules or reference to other files that contain Snort rules. In addition to rules, the configuration file also contains information about input and output plug-ins. The typical name of the Snort configuration file is snort.conf. The following command starts Snort in the Network Intrusion Detection (NID) mode [115].

```
snort -c /opt/snort/etc/snort.conf
```

When this command is started snort will read the configuration file snort.conf and all other files included in this file. Typically these files contain Snort rules and configuration data. After reading these files, Snort will build its internal data structures and rule chains. All captured packets will then be matched against these rules and appropriate action will be taken, if configured to do so.

Network Sniffer Mode:

In the network sniffer mode, Snort acts like the commonly used program tcpdump. It can capture and display packets from the network with different levels of detail on the console. You don’t need a configuration file to run Snort in the packet sniffing mode. The following command displays information about each packet flowing on the network segment [115]:

```
/snort –v
```

Snort will continue to display captured packets on the screen until you break using Ctrl-C. At the time Snort terminates, it will display statistical information. The following is a typical output for a TCP packet [122]: If you analyze the output in figure 4.1 the following information about the packet can be viewed:

- Date and time the packet was captured 13th of April at 5:58:16.
- Source IP address is 192.168.58131.
- Source port number is 22.
- Destination IP address is 192.168.58.242.
- Transport layer protocol used in this packet is UDP.
- Time To Live or TTL value in the IP header part is 64.
- Type of Service or TOS value is 0x0.
- Packet ID is 38864.
- Length of IP header is 20.
- IP payload is 207 bytes long.
- Don’t Fragment or DF length is 235

3.2.2. Iptable firewall

Iptables is part of the Netfilter project. Netfilter is a set of Linux kernel hooks that communicate with the network stack. Iptables is a command and the table structure that contains the rule sets that control the packet filtering [70]. Iptables are complex, it filters packets by the fields in IP, UDP, and ICMP packet headers and a number of different actions can be taken on each packet.

The iptables architecture groups network packet processing rules into tables by function (packet filtering, NAT, and other packet mangling), each of which have chains (sequences) of processing rules. Rules consist of matches (used to determine which packets the rule will apply to) and targets (that determine what will be done with the matching packets). Iptables operates at OSI Layer 3 (Network). For OSI Layer 2 (Link), there are other technologies such as ebtables (Ethernet Bridge Tables) [123]. Presently, this firewall has become more and more popular (both among end users and network administrators). The popularity of this firewall is closely related to Linux operating system, because iptables works with Linux kernels 2.4 and 2.6 and almost every major Linux distribution comes with pre-installed iptables firewall [124].

The core of this firewall consists of four parts; tables, chains, matches and targets [124]. A system administrator is able to define an iptables policy, i.e. tables of chains, which describe how a kernel should react against different groups of packets. On the other hand,[70] states that there three tables in iptables, any rules or custom chains that are created will always go into one of these tables. The filter table is the default, and is the one most used. The filter table contains five predefined chains: INPUT, OUTPUT, FORWARD, and PREROUTING AND POSTROUTING.

PREROUTING is used for altering packets just as they enter the firewall and before they hit the routing decision. POSTROUTING is used to mangle packets just after all routing decisions have been made. OUTPUT is used for altering locally generated packets before they enter the routing decision. INPUT is used to alter packets after they have been routed to the local computer itself, but before the user space application actually sees the data. FORWARD is used to mangle packets after they have hit the first routing decision, but before they actually hit the last routing decision. Note that mangle cannot be used for any kind of Network Address Translation or Masquerading; the nat table was made for these kinds of operations [115].

The **filter** table should be used exclusively for filtering packets. For example, we could **DROP**, **LOG**, **ACCEPT** or **REJECT** packets without problems, as we can in the other tables. There are three chains built in to this table. The first one is named FORWARD and is used on all non-locally generated packets that are not destined for local host (the firewall). INPUT is used on all packets that are destined for our local host (the firewall) and OUTPUT is finally used for all locally generated packets. When a packet first enters the firewall, it hits the hardware and then get has passed on to the proper device driver in the kernel.

Then the packet starts to go through a series of steps in the kernel before it is either sent to the correct application (locally), or forwarded to another host or whatever happens to it. The packet goes through the different steps Figure 2.12.1 explains how packets traverse the different chains and in which order [115]. When a packet first enters the firewall, it hits the hardware and then get has passed on to the proper device driver in the kernel. Then the packet starts to go through a series of steps in the kernel before it is either sent to the correct application (locally), or forwarded to another host or whatever happens to it. The packet goes through the different steps (figure 3.4 below) explains how packets traverse the different chains and in which order.

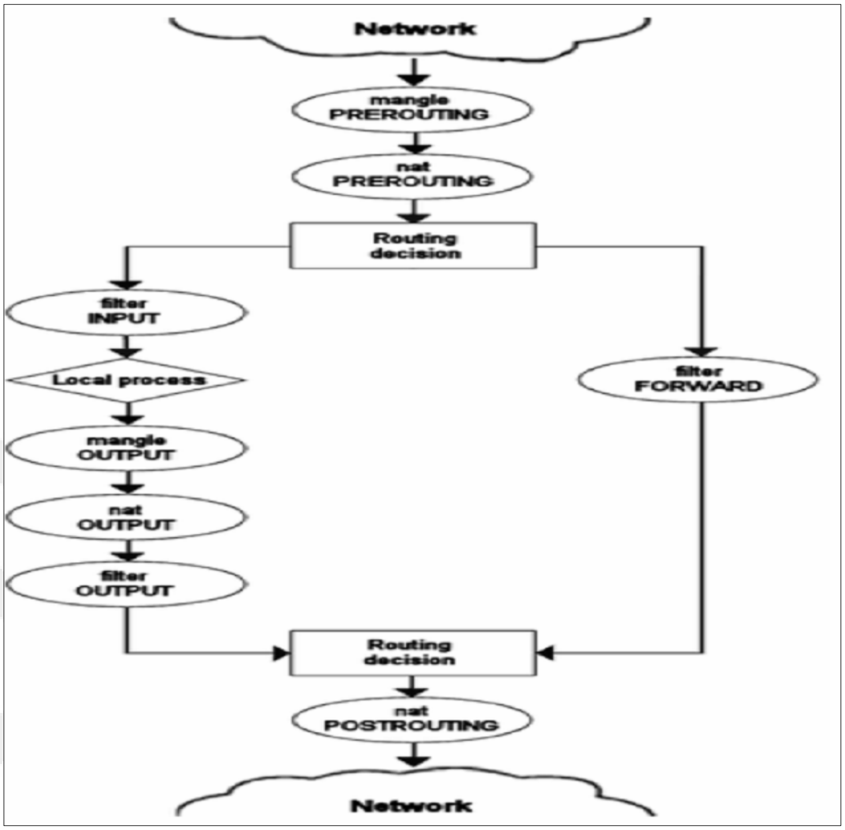


Figure 3.4. Shows how the packets traverse the built in chains now (Asarcıklı 2005).

3.2.2.1. How to write rules

Iptables are used to create a chain collection of rules that are applied to every packet [124]. Each rule is a line that the kernel looks at to find out what to do with a packet. If all the criteria or matches are met, the target instruction is performed [115]. And normally the rules are written using the syntax below;

Iptables (-t table) command (match) (target/jump)

The **-t** option specifies which table to use. Per default, the filter table is used. We may specify one of the following tables with the -t option. The **nat** table is used mainly for Network Address Translation. "NAT"ed packets get their IP addresses altered, according to rules. Packets in a stream only traverse this table once. It is assumed that the first packet of a stream is allowed. The rest of the packets in the same stream are automatically "NAT"ed or Masqueraded etc, and will be subject to the same actions as the first packet. These will not go through this table again, but will nevertheless be treated like the first packet in the stream. This is the main reason why you should not do any filtering in this table. The **mangle** table is used mainly for mangling packets. Among other things, we can change the contents of different packets and that of their headers. Examples of this would be to change the **TTL**, **TOS** or **MARK**. The **MARK** is not really a change to the packet, but a mark value for the packet is set in kernel space.

The command should always come first, or alternatively directly after the table specification. The “command” is used to tell the program what to do, for example to insert a rule or to add a rule to the end of the chain, or to delete a rule. The “match” is the part of the rule that is sent to the kernel that details the specific character of the packet, what makes it different from all other packets. Here we can specify what IP address the packet comes from, from which network interface, the intended IP address, port, protocol or whatever.

Finally, the packet has the target. If all the matches are met for a packet, “target/jump” part of the command tells the kernel what to do with it. We could, for example, tell the kernel to send the packet to another chain that we have created ourselves, and which is part of this particular table. We could tell the kernel to drop the packet dead and do no further processing, or we could tell the kernel to send a specified reply to the sender.

A chain is just a simple checklist of rules and specifies what to do with each of the packets. The chain rule will either ACCEPT a packet or DROP a packet. If the packet does not have any more rules left in the chain, the system will consult the chain policy to decide what to do. Most systems are setup with a policy of deny. Therefore, if the packet does not match any rules that "allow" it through, then it will "drop" it. Iptables uses a set of chain rules. The three default chains are named INPUT, OUTPUT, and FORWARD. A chain is just a simple checklist of rules and specifies what to do with each of the packets. The chain rules will either ACCEPT a packet or DROP a packet. If the packet does not have any more rules left in the chain, the system will consult the chain policy to decide what to do. Most systems are setup with a policy of deny. Therefore, if the packet does not match any rules that "allow" it through, then it will "drop" it [115]

The first decision the kernel has to make upon receipt of a packet is "Where is the destination?" If the destination is for the box itself, it will consult the rules for the INPUT chain. If the destination is for another network interface (and IP Forwarding is enabled), the packet is compared against the FORWARD chain. As long as the packet gets an "ACCEPT" by one of the chain rules the packet will be forwarded on. If the Linux box itself needs to send network packets, it will consult the OUTPUT chain and if the packet is ACCEPTED by one of the rules, it will be sent out to the appropriate interface. One of the key concepts is that the INPUT and OUTPUT chains actually refer to the local machine rather than to all incoming and outgoing packets [115].

The three built-in chains are INPUT, OUTPUT and FORWARD and they cannot be deleted. The followings are operations to manipulate these chains; [113, 115],

- Create a new chain (-N).
- Delete an empty chain (-X).
- Change the policy for a built-in chain. (-P).
- List the rules in a chain (-L).
- Flush the rules out of a chain (-F).
- Zero the packet and byte counters on all rules in a chain (-Z).

There are several ways to manipulate rules inside a chain:

- Append a new rule to a chain (-A).
- Insert a new rule at some position in a chain (-I).

- Replace a rule at some position in a chain (-R).
- Delete a rule at some position in a chain, or the first that matches (-D).

Some other options:

- **j** Specify the target (--jump)
- **i** Specify the input interface (--in-interface)
- **o** Specify the output interface (--out-interface)
- **p** Specify the protocol (--protocol)
- **s** Specify the source (--source)
- **d** Specify the destination (--destination)
- **!** Specifies an inversion (match addresses NOT equal to)

3.2. ARPwatch

ARPwatch is an open source computer software program developed by Lawrence Berkeley National Laboratory, a Network Research Group and was released under the BSD license. ARPwatch helps to monitor Ethernet traffic activity (like changing IP and MAC address) on your network and maintains a database of ethernet/IP address pairings. It produces a log of notice pairing of IP and MAC address information along with a timestamps, so carefully when a person tries to watch when the pairing activity appeared on the network. It also has the option to send reports via email to a network administrator when pairing added or changed [125].

ARPwatch is a free UNIX program which listens for ARP replies on a network. It will build a table of IP/MAC associations and store them in a file. When the MAC address associated with an IP changes (referred to as a flip-flop), an email is sent to an administrator. Network administrators monitor ARP activity to detect ARP spoofing.

CHAPTER 4. RESULTS AND DISCUSSIONS

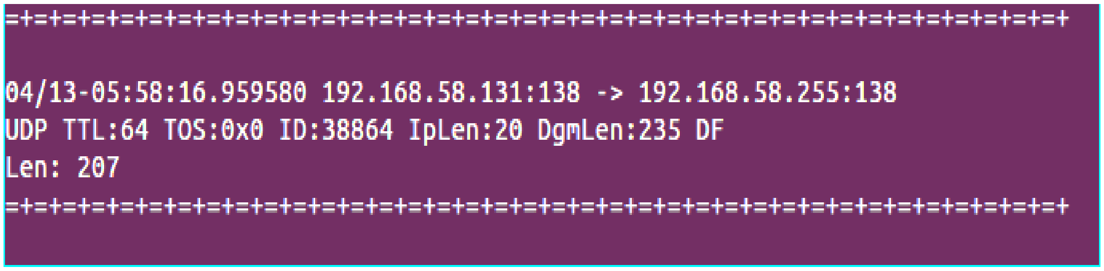
This chapter explains the results obtained during the study which include among others results from sniffing, detection, prevention.

4.1. Sniffing

Before running the snort rules, it was vital to do sniffing to the network so that to check whether there were anomalies in the data packets. Snort was made to run in the network sniffer mode. Here a configuration file to run snort in sniffing mode was not needed. By running the command below snort was made to run in sniffing mode. The command displays information about each packet flowing on the network segment:

```
./snort -v
```

Snort acts like the commonly used program tcpdump. It captured and displayed packets from the network with different levels of detail on the console. Consecutively, snort displayed and captured packets on the screen showed in figure 4.1.



```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/13-05:58:16.959580 192.168.58.131:138 -> 192.168.58.255:138
UDP TTL:64 TOS:0x0 ID:38864 IpLen:20 DgmLen:235 DF
Len: 207
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

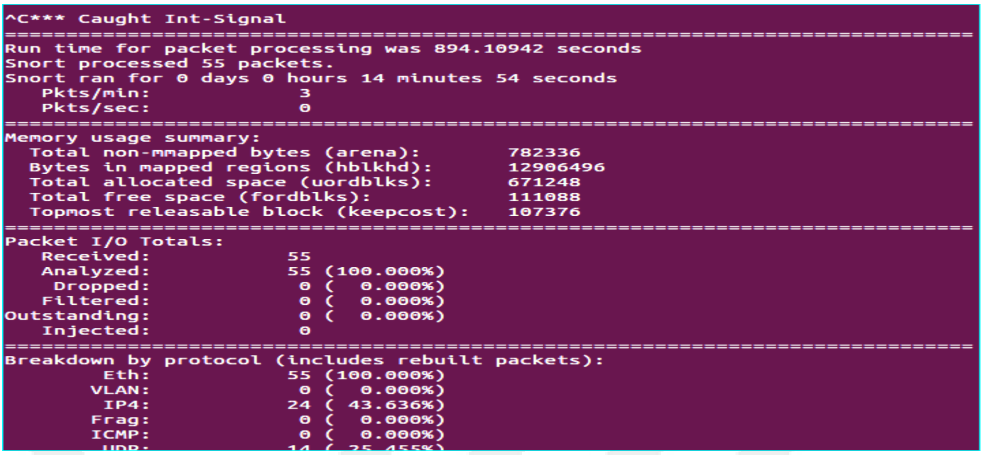
Figure 4.1. Sniff data output.

To analyze the output in (figure 24) the following information about the packet was captured;

- Date and time the packet was captured 13th of April at 5:58:16.
- Source IP address is 192.168.58131.
- Source port number is 22.
- Destination IP address is 192.168.58.242.
- Transport layer protocol used in this packet is UDP.
- Time to Live or TTL value in the IP header part is 64.
- Type of Service or TOS value is 0x0.
- Packet ID is 38864.
- Length of IP header is 20.
- IP payload is 207 bytes long.
- Don't Fragment or DF length is 235.

And to break the sniffing process Ctrl-C was used and at the same time snort was terminated, statistical information was displayed. This section provides basic Statistics;

- Protocol statistics; Traffic for all the protocols decoded by Snort is summarized in the breakdown section.
- Packet I/O Totals; This section shows basic packet acquisition and injection peg counts obtained from the DAQ. Includes received, analyzed, Dropped, Filtered, injected and Outstanding packets is displayed.
- Timing statistics; it includes total seconds and packets as well as packet processing rates. The rates are based on whole seconds, minutes, etc. and only shown when non-zero.



```
^C*** Caught Int-Signal
===============================================================================
Run time for packet processing was 894.10942 seconds
Snort processed 55 packets.
Snort ran for 0 days 0 hours 14 minutes 54 seconds
   Pkts/min:             3
   Pkts/sec:             0
===============================================================================
Memory usage summary:
  Total non-mmapped bytes (arena):       782336
  Bytes in mapped regions (hblkhd):      12906496
  Total allocated space (uordblks):      671248
  Total free space (fordblks):           111088
  Topmost releasable block (keepcost):   107376
===============================================================================
Packet I/O Totals:
   Received:            55
   Analyzed:            55 (100.000%)
    Dropped:             0 (  0.000%)
   Filtered:             0 (  0.000%)
Outstanding:             0 (  0.000%)
   Injected:             0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:            55 (100.000%)
       VLAN:             0 (  0.000%)
        IP4:            24 ( 43.636%)
       Frag:             0 (  0.000%)
       ICMP:             0 (  0.000%)
        UDP:            14 ( 25.455%)
```

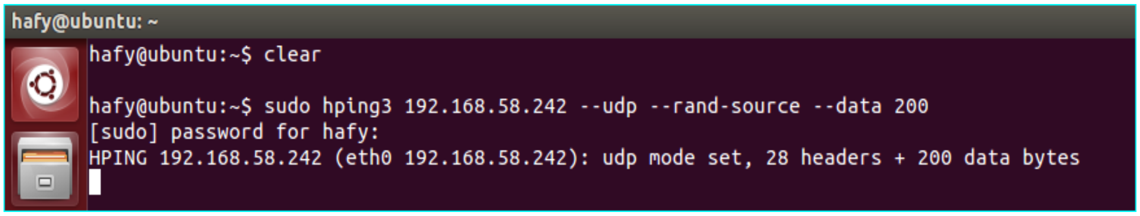
.

Figure 4.2. Statistical information captured after sniffing.

Next was detection of attacks on the network. Our study's case focused only on denial of service attacks (DOS) and ARP spoofing attacks. Thus for this case different modules were experimented, modules for DOS attacks and another for ARP spoofing attack.

Denial of service (DOS) module, experiments setups were carried using Linux and Ubuntu version 14.0 machine and installed Wireshark for network protocol analyzing. Here different DOS attacks were carried for example; a SYN-Flood attack, ICMP Flood attack and UDP flood attack modules using hping3 tool.

UDP-Flood attack; the attack was made by running the following hping3 command from the attackers machine, and this led to Flooding the victim's machine.
For UDP flood attack, the following hping3 command was run.



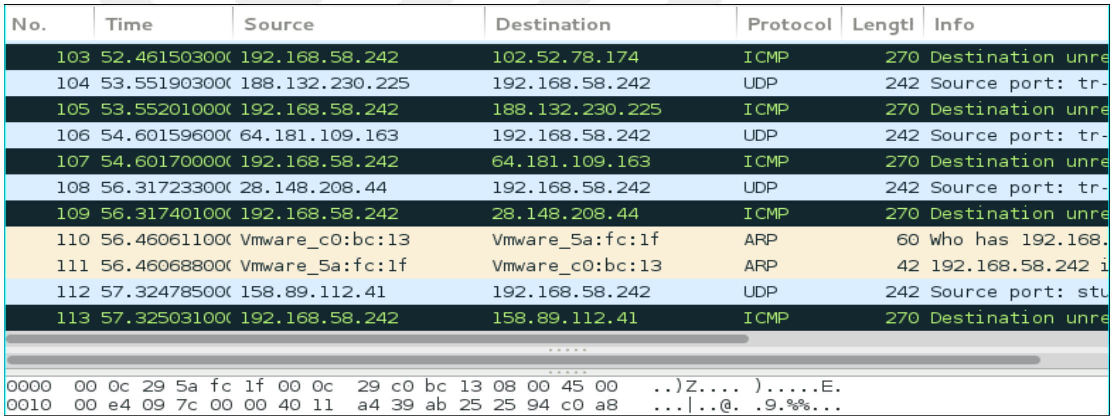
```
hafy@ubuntu:~$ clear

hafy@ubuntu:~$ sudo hping3 192.168.58.242 --udp --rand-source --data 200
[sudo] password for hafy:
HPING 192.168.58.242 (eth0 192.168.58.242): udp mode set, 28 headers + 200 data bytes
```

Figure 4.3. Udp hping3 script

Description;

- 192.168.58.242 is a targeted IP
- udp flag set the udp mode.
- p 80 sets packets to port 80 on victim’s machine (192.168.58.242)
- data 200 set the interval between packets as 200 packets per second.
- rand-source is used for random source IP addresses,

So when Wireshark is run, random IP addresses are captured as a source IP addresses and destination IP as shown in figure.



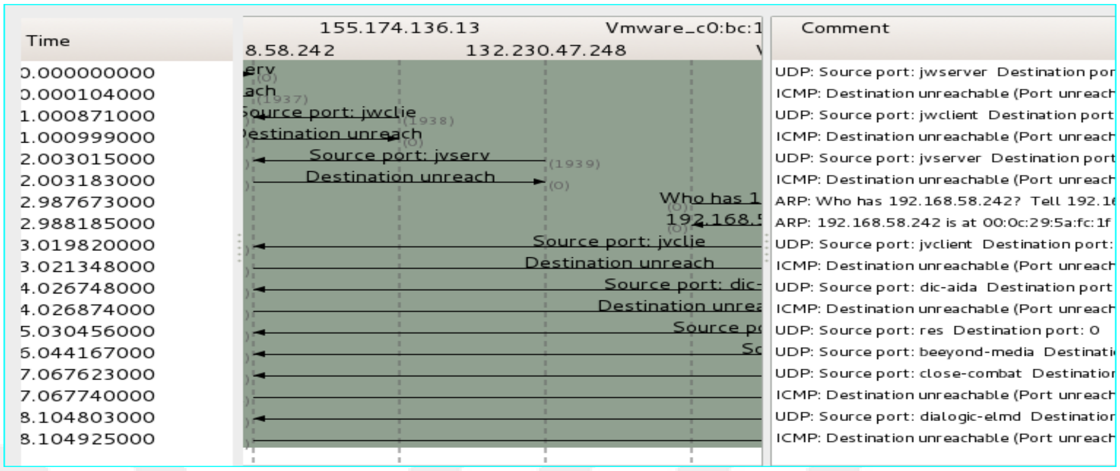
No.	Time	Source	Destination	Protocol	Lengtl	Info
103	52.46150300(	192.168.58.242	102.52.78.174	ICMP	270	Destination unre
104	53.55190300(	188.132.230.225	192.168.58.242	UDP	242	Source port: tr-
105	53.55201000(	192.168.58.242	188.132.230.225	ICMP	270	Destination unre
106	54.60159600(	64.181.109.163	192.168.58.242	UDP	242	Source port: tr-
107	54.60170000(	192.168.58.242	64.181.109.163	ICMP	270	Destination unre
108	56.31723300(	28.148.208.44	192.168.58.242	UDP	242	Source port: tr-
109	56.31740100(	192.168.58.242	28.148.208.44	ICMP	270	Destination unre
110	56.46061100(	Vmware_c0:bc:13	Vmware_5a:fc:1f	ARP	60	Who has 192.168.
111	56.46068800(	Vmware_5a:fc:1f	Vmware_c0:bc:13	ARP	42	192.168.58.242 i
112	57.32478500(	158.89.112.41	192.168.58.242	UDP	242	Source port: stu
113	57.32503100(	192.168.58.242	158.89.112.41	ICMP	270	Destination unre

```
0000   00 0c 29 5a fc 1f 00 0c   29 c0 bc 13 08 00 45 00   ..)Z.... ).....E.
0010   00 e4 09 7c 00 00 40 11   a4 39 ab 25 25 94 c0 a8   ...|..@. .9.%%...
```

Figure 4.4. Captured information using wireshark.

Figure 4.3. Below Shows that the victim’s machine (192.168.58.242) was responding with ICMP port unreachable since there was no application running on attacker’s machine which sent UDP packet. And this indicated that the host does not exist.

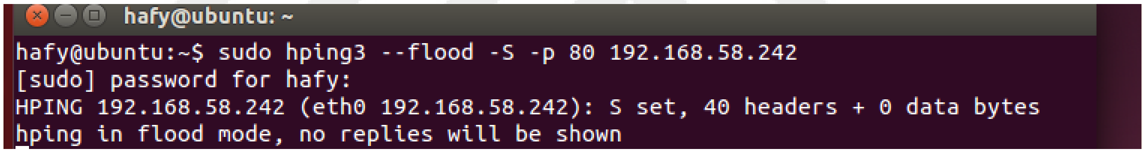
In this way, all of the resources of the victim’s machine were consumed and this made legitimate requests not to be served since the victim was busy in serving attacker’s invalid requests.



Time	Comment
0.000000000	UDP: Source port: jwserver Destination por
0.000104000	ICMP: Destination unreachable (Port unreach
1.000871000	UDP: Source port: jwclient Destination port
1.000999000	ICMP: Destination unreachable (Port unreach
2.003015000	UDP: Source port: jvserver Destination port
2.003183000	ICMP: Destination unreachable (Port unreach
2.987673000	ARP: Who has 192.168.58.242? Tell 192.16
2.988185000	ARP: 192.168.58.242 is at 00:0c:29:5a:fc:1f
3.019820000	UDP: Source port: jvclient Destination port:
3.021348000	ICMP: Destination unreachable (Port unreach
4.026748000	UDP: Source port: dic-aida Destination port
4.026874000	ICMP: Destination unreachable (Port unreach
5.030456000	UDP: Source port: res Destination port: 0
6.044167000	UDP: Source port: beeyond-media Destinati
7.067623000	UDP: Source port: close-combat Destination
7.067740000	ICMP: Destination unreachable (Port unreach
8.104803000	UDP: Source port: dialogic-elmd Destination
8.104925000	ICMP: Destination unreachable (Port unreach

Figure 4.5. Shows a flow graph after hping3 script is run and recapture packets using Wireshark

For SYN flood attack, the following hping3 command was run;

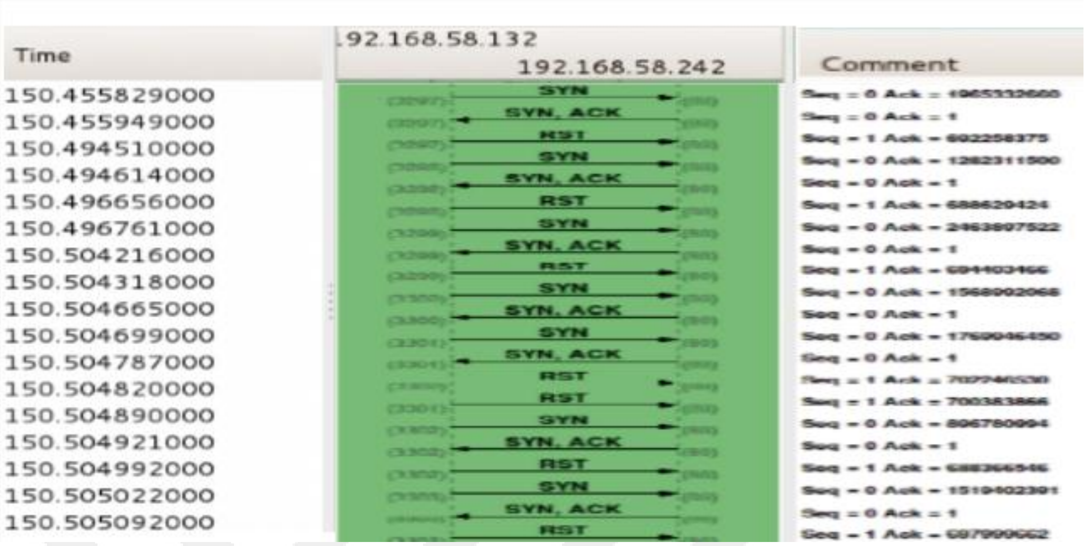


```
hafy@ubuntu:~$ sudo hping3 --flood -S -p 80 192.168.58.242
[sudo] password for hafy:
HPING 192.168.58.242 (eth0 192.168.58.242): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Explanation of the command;

– flood flag sent the packets as fast as possible
– S flag set the SYN flag on in TCP mode
– p 80 sent the packet to port 80 on victim’s machine 192.168.58.242

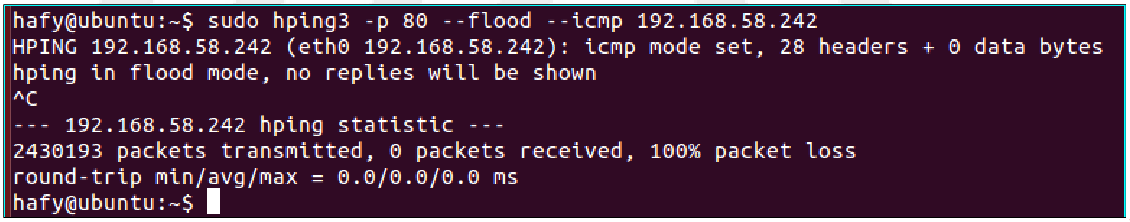
On a victim machine, capturing and analyzing traffic using Wireshark show that victim machine (192.168.58.132) is responding to SYN packet by sending back packets with SYN, ACK flags set, but the attacker's machine (192.168.58.242) is not participating three way handshake by sending back ACK, instead it is sending RST flag set packet thereby resulting in half open connection. And when thousands of such connections are made in a few seconds, eventually the victim's machine will get exhausted in no time. Fig. 4.6 shows TCP flow graph using SYN flood attack.



```
Time                 .92.168.58.132
                             192.168.58.242      Comment
150.455829000                 SYN              Seq = 0 Ack = 1965332600
150.455949000               SYN, ACK           Seq = 0 Ack = 1
150.494510000                 RST              Seq = 1 Ack = 602258375
150.494614000                 SYN              Seq = 0 Ack = 1282311500
150.496656000               SYN, ACK           Seq = 0 Ack = 1
150.496761000                 RST              Seq = 1 Ack = 688620424
150.504216000                 SYN              Seq = 0 Ack = 2463807522
150.504318000               SYN, ACK           Seq = 0 Ack = 1
150.504665000                 RST              Seq = 1 Ack = 691403466
150.504699000                 SYN              Seq = 0 Ack = 1568002066
150.504787000               SYN, ACK           Seq = 0 Ack = 1
150.504820000                 SYN              Seq = 0 Ack = 1769046450
150.504890000               SYN, ACK           Seq = 0 Ack = 1
150.504921000                 RST              Seq = 1 Ack = 702746530
150.504992000                 RST              Seq = 1 Ack = 700383866
150.505022000                 SYN              Seq = 0 Ack = 896780094
150.505092000               SYN, ACK           Seq = 0 Ack = 1
                              RST              Seq = 1 Ack = 688366546
                              SYN              Seq = 0 Ack = 1519402301
                            SYN, ACK           Seq = 0 Ack = 1
                              RST              Seq = 1 Ack = 697900062
```

Figure 4.6. Shows a flow graph after hping3 script is run and recapture packets using wireshark.

ICMP flood attack module was also experimented by running the hping3 command below;



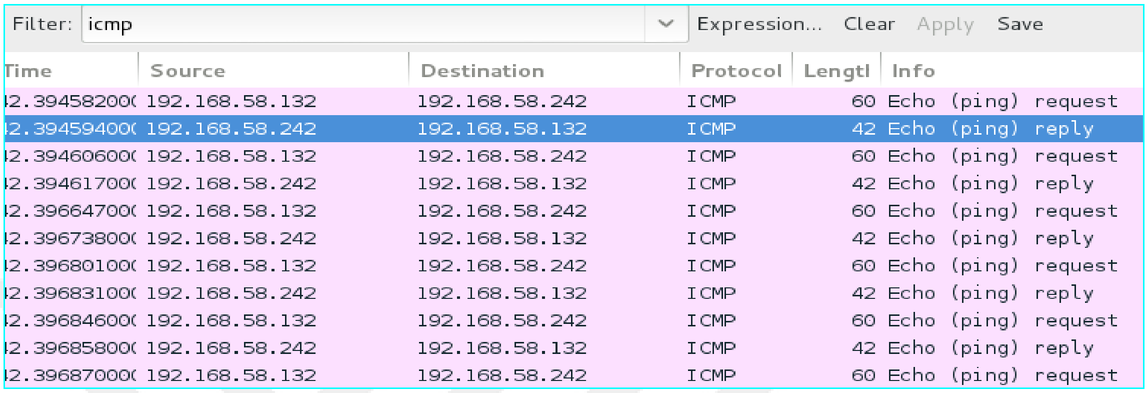
```
hafy@ubuntu:~$ sudo hping3 -p 80 --flood --icmp 192.168.58.242
HPING 192.168.58.242 (eth0 192.168.58.242): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.58.242 hping statistic ---
2430193 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
hafy@ubuntu:~$ 
```

Figure 4.7. Icmp hping3 script

Description of the of the script

- P 80 sent packets to port 80 on victim’s machine (192.168.58.242)
- flood flag sent the packet as fast as possible
- icmp flag set the icmp mode

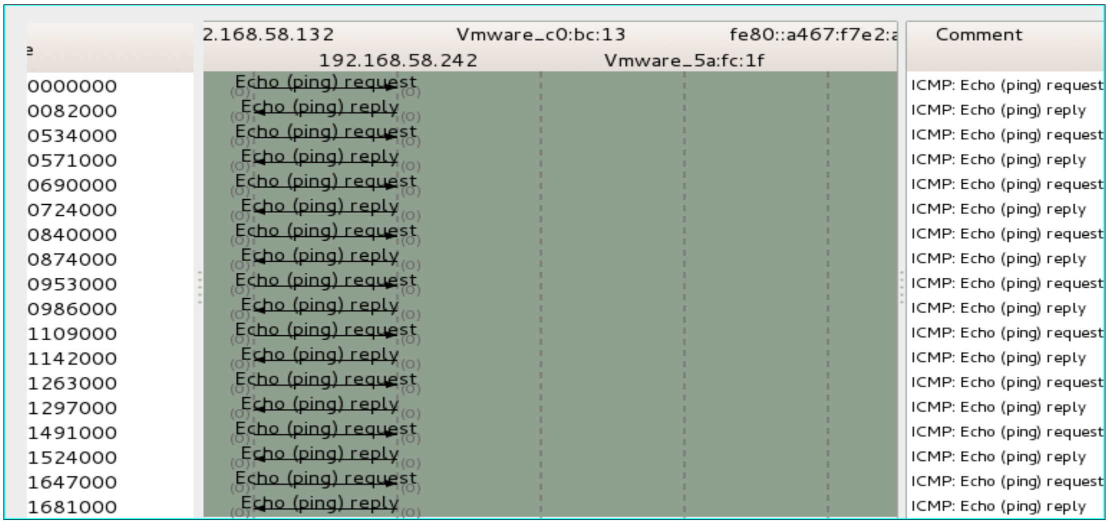
In figure 4.8 an attacker sends large number of ICMP Echo Request messages to the targeted machine (192.168.58.242)



Time	Source	Destination	Protocol	Lengtl	Info
12.39458200(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request
12.39459400(	192.168.58.242	192.168.58.132	ICMP	42	Echo (ping) reply
12.39460600(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request
12.39461700(	192.168.58.242	192.168.58.132	ICMP	42	Echo (ping) reply
12.39664700(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request
12.39673800(	192.168.58.242	192.168.58.132	ICMP	42	Echo (ping) reply
12.39680100(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request
12.39683100(	192.168.58.242	192.168.58.132	ICMP	42	Echo (ping) reply
12.39684600(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request
12.39685800(	192.168.58.242	192.168.58.132	ICMP	42	Echo (ping) reply
12.39687000(	192.168.58.132	192.168.58.242	ICMP	60	Echo (ping) request

Figure 4.8. Captured information using wireshark

In figure 4.5 below, the attacker machine (192.168.58.242) sent a lot of icmp requests to the victim's machine (192.168.58.131) simultaneously and the victim machine could give back reply to the requests made by the attacking machine. This made the target machine to typically get flooded with ICMP echo requests thus overloading it with a lot of requests which led to degrading the performance of the target machine thus leading to a denial of service attack.



e		Comment
0000000	Echo (ping) request	ICMP: Echo (ping) request
0082000	Echo (ping) reply	ICMP: Echo (ping) reply
0534000	Echo (ping) request	ICMP: Echo (ping) request
0571000	Echo (ping) reply	ICMP: Echo (ping) reply
0690000	Echo (ping) request	ICMP: Echo (ping) request
0724000	Echo (ping) reply	ICMP: Echo (ping) reply
0840000	Echo (ping) request	ICMP: Echo (ping) request
0874000	Echo (ping) reply	ICMP: Echo (ping) reply
0953000	Echo (ping) request	ICMP: Echo (ping) request
0986000	Echo (ping) reply	ICMP: Echo (ping) reply
1109000	Echo (ping) request	ICMP: Echo (ping) request
1142000	Echo (ping) reply	ICMP: Echo (ping) reply
1263000	Echo (ping) request	ICMP: Echo (ping) request
1297000	Echo (ping) reply	ICMP: Echo (ping) reply
1491000	Echo (ping) request	ICMP: Echo (ping) request
1524000	Echo (ping) reply	ICMP: Echo (ping) reply
1647000	Echo (ping) request	ICMP: Echo (ping) request
1681000	Echo (ping) reply	ICMP: Echo (ping) reply

Figure 4.9. Shows victim's machine under ICMP flood attack.

For ARP spoofing attack, an experiement was done too using ettercap. The first step was to set an IP address on the the machine with ettercap to be in the same subnet.
Firstly, the network interfaces was configured using the command below;

```
#ifconfig interface_name ip_address, netmask mask

Ifconfig eth0 192.168.58.242 netmask 255.255.255.0 broadcast
```

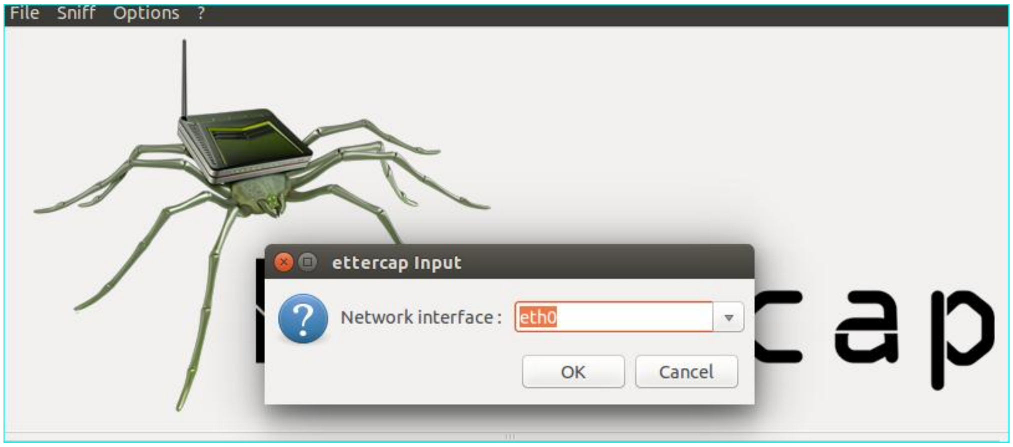
To configure the default gateway:

```
#route add default gateway ip_address dev interface_name

#route add default gateway 192.168.1.1 dev eth0
```

To launch Ettercap so that to start the ARP spoofing, the following command was run on the attacker’s machine and it ordered ettercap to be opened in a graphical mode.

```
#ettercap -G
```

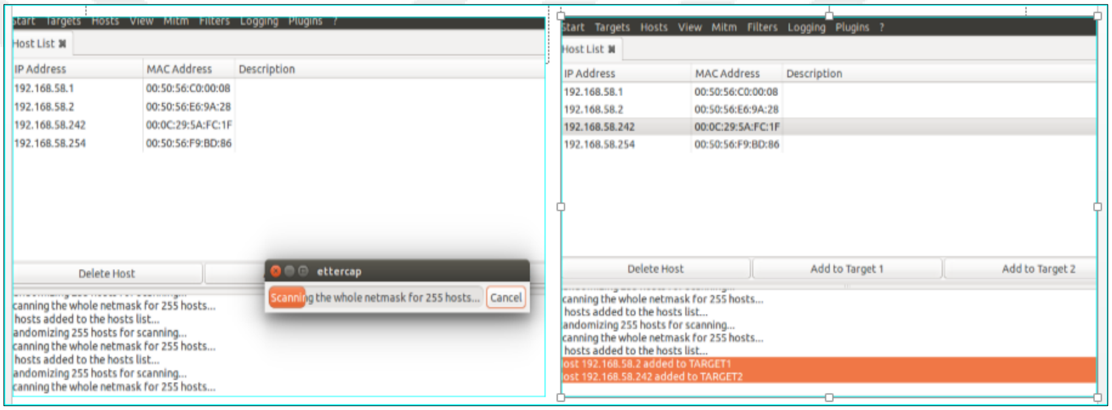
From there, ettercap was configured ettercap for unified sniffing. The interface (eth0) was selected to sniff on Ubuntu attacker machine.



Network interface : eth0

Figure 4.10. Network Interface to sniff.

Later the hosts inside the subnet were scanned so that to find the target machines. The network range scanned was determined by the IP settings of the interface chosen in the previous step. And 4 hosts were found in the host list. Then the target machines were selected. The target machines to ARP spoofing were Linux machine (192.168.58.242) and the gateway (192.168.58.2). Specified the targets by highlighting the line containing 192.168.58.2 and clicked on the “target 1” button. Further, we highlighted the line containing 192.168.58.242 and clicked on the “target 2” button.


Start Targets Hosts View Mitm Filters Logging Plugins ?
Host List
Delete Host
ettercap
Cancel
Start Targets Hosts View Mitm Filters Logging Plugins ?
Host List
Delete Host
Add to Target 1
Add to Target 2

IP Address	MAC Address	Description
192.168.58.1	00:50:56:C0:00:08	
192.168.58.2	00:50:56:E6:9A:28	
192.168.58.242	00:0C:29:5A:FC:1F	
192.168.58.254	00:50:56:F9:BD:86	

```
canning the whole netmask for 255 hosts...
 hosts added to the hosts list...
andomizing 255 hosts for scanning...
canning the whole netmask for 255 hosts...
 hosts added to the hosts list...
andomizing 255 hosts for scanning...
canning the whole netmask for 255 hosts...
```

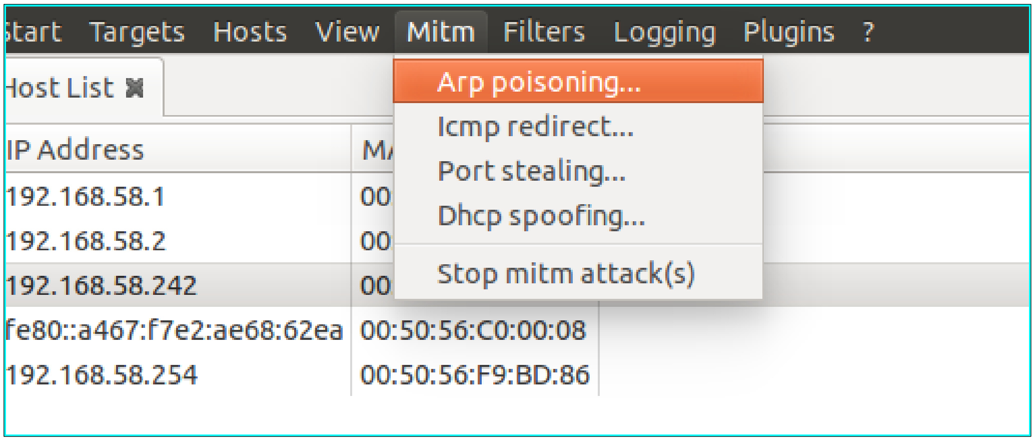
Scanning the whole netmask for 255 hosts...

IP Address	MAC Address	Description
192.168.58.1	00:50:56:C0:00:08	
192.168.58.2	00:50:56:E6:9A:28	
192.168.58.242	00:0C:29:5A:FC:1F	
192.168.58.254	00:50:56:F9:BD:86	

```
canning the whole netmask for 255 hosts...
 hosts added to the hosts list...
andomizing 255 hosts for scanning...
canning the whole netmask for 255 hosts...
 hosts added to the hosts list...
ost 192.168.58.2 added to TARGET1
ost 192.168.58.242 added to TARGET2
```

Figure 4.11. Shows scanning of the hosts in the subnet and the MAC and IPs of the target machines to be sniffed.

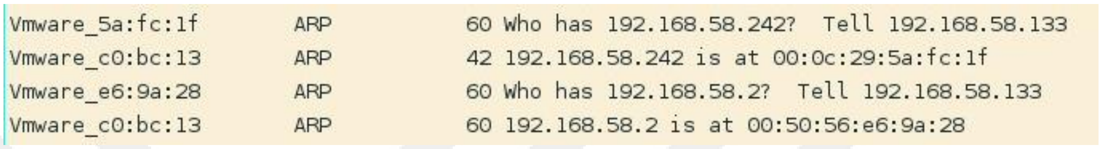
Then started the MITM (ARP poisoning).



IP Address	M/
192.168.58.1	00
192.168.58.2	00
192.168.58.242	00
fe80::a467:f7e2:ae68:62ea	00:50:56:C0:00:08
192.168.58.254	00:50:56:F9:BD:86

Figure 4.12. Shows ARP spoofing.

By running Wireshark on the Linux machine it was getting ARP broadcast messages and it could give reply (unicast). Here an ARP request was broadcast to all machines on the network asking which machines had those targeted IP address and MAC and the machines responded with a unicast replies telling the attacker machines the MAC and IP addresses for the requested machines.



```
Vmware_5a:fc:1f       ARP          60 Who has 192.168.58.242?  Tell 192.168.58.133
Vmware_c0:bc:13       ARP          42 192.168.58.242 is at 00:0c:29:5a:fc:1f
Vmware_e6:9a:28       ARP          60 Who has 192.168.58.2?  Tell 192.168.58.133
Vmware_c0:bc:13       ARP          60 192.168.58.2 is at 00:50:56:e6:9a:28
```

Figure 4.13. An Ettercap ARP Spoofing attack as seen by wireshark.

With the help of Wireshark, the ARP traffic for before and after the attack was compared. Before the attack, the gateway and Linux machine to be able to communicate together they sent an ARP broadcast to find out the MAC addresses of the other. And after spoofing the ARP broadcast request was answered by the gateway.

4.2. Detection

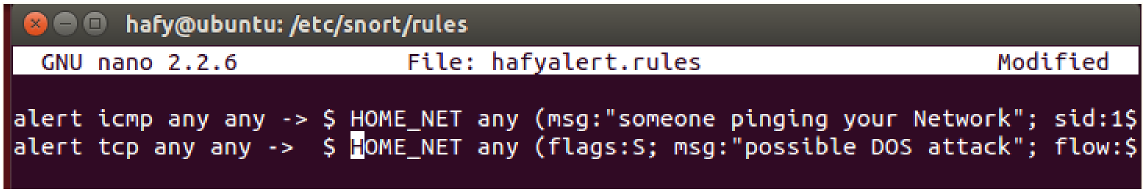
An IDS was used to detect attacks on the network. The Intrusion Detection System was placed at the entry of the network (before) the firewall so that it monitor traffic flows from internal network to the external network or from external network to internal network and for this case, SNORT was used.
First, snort rules were written in the

/etc/snort/rules

Using the command below Snort rules were written so that to detect attacks on the network as shown in figure.

#action protocol src_IP src_port Direction dst_IP dst_port (msg: “A message”; option rev: number: more optional options)

The following were the snort rules written to detect denial of service attacks.



```
  GNU nano 2.2.6              File: hafyalert.rules                       Modified

alert icmp any any -> $ HOME_NET any (msg:"someone pinging your Network"; sid:1$
alert tcp any any ->  $ HOME_NET any (flags:S; msg:"possible DOS attack"; flow:$
```

Figure 4.14. Snort rules

Rule1: Alert icmp any any -> any any (msg:"Someone pinging your network"; sid:100001) This rule means that for any ICMP ping packets that it sees from any network to the HOME_NET, generate an alert with the text or message “someone pinging your network”.

Rule 2: Means that whenever a DOS attack is done from any network to the HOME_NET, an alert with the text message of “DDOS flood denial of service attempt”.

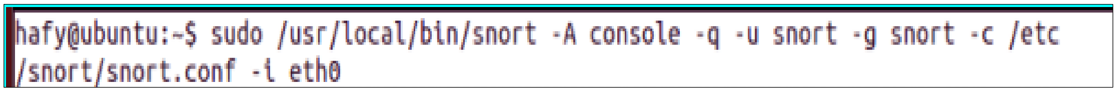
After the snort data passes through the detection engine, an whenever data could not match a rule in the detection engine, an alert was made. Depending on what the detection could find inside the packet, the packet was used to logg the activity or generate an alert. And to do this, the following were done;Since it is in the local rules.file where rules that are specifically for generating alerts re placed. So first step was to edit the local.rules file so that to place the rules there by using the following command;

```
#sudo vi  /etc/snort/rules/local.rules
```

Later the rules were input in the local.files folder and were saved using Ctrl C. And since changes were made to the files that snort loads, it was important to test the configuration file again to check whether the snort configurations are still in order. Using the following command;

```
# sudo snort –T –c /etc/snort/snort.conf
```

Next snort was run to detect and generate an alert using the following command;

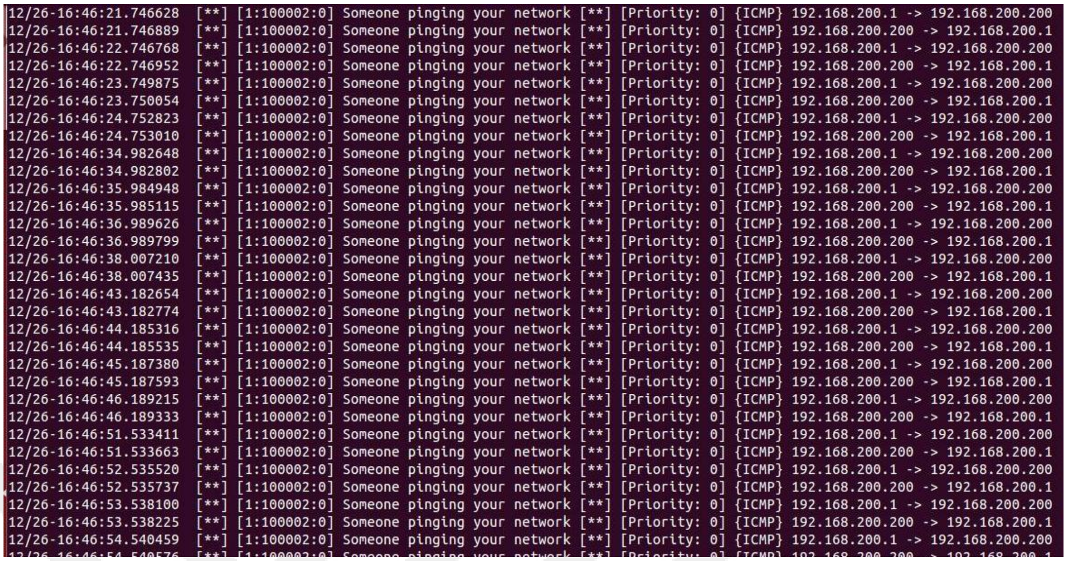


```
hafy@ubuntu:~$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc
/snort/snort.conf -i eth0
```

- A console; indicates that the warnings are to be displayed on the terminal console.
- q; quiet option, and it does not show the banner and status report
- u snort; specified to run snort as the following user after startup.
- g snort; specifies to run snort as the following group after startup.
- c; indicates the path to snort.conf file.
- eth0; indicates the network interface to listen to.

To detect the attack, denial of service attacks were launched as explained at earlier at the beginining of this chapter. The IDS were used in a way that whenever any attack was launched on the network, the detection engine could send an alert to the network administrator. And this was inform of an alarm or alert message in order to inform system administrators that there is an attack launched on the network.

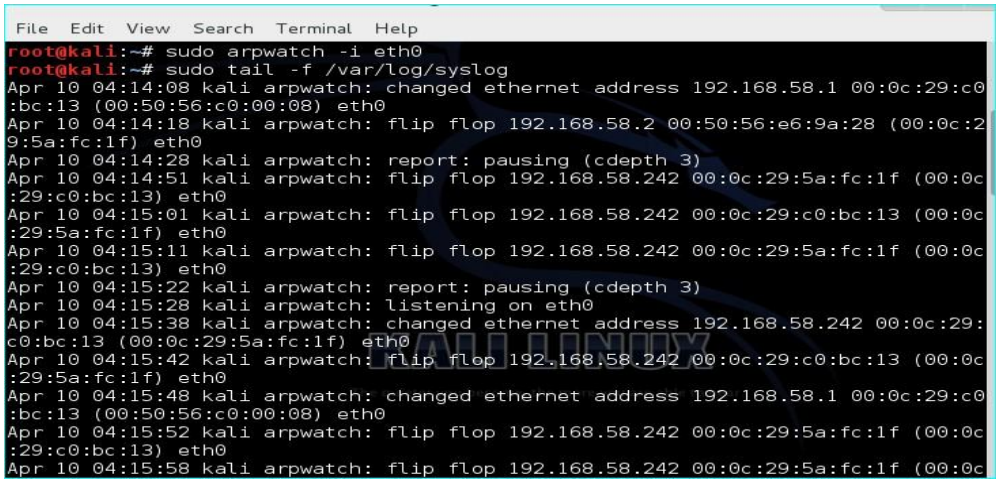
The alert triggered was on screen alert and it was displayed on the IDS console since this specification was given when configuring the IDS. The information provided in the alert message varies widely, ranging from a notification that an intrusion has taken place to extremely detailed messages outlining the IP addresses of the source and target of the attack, attack type, the time at which the attack occurred.



```
12/26-16:46:21.746628  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:21.746889  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:22.746768  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:22.746952  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:23.749875  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:23.750054  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:24.752823  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:24.753010  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:34.982648  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:34.982802  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:35.984948  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:35.985115  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:36.989626  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:36.989799  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:38.007210  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:38.007435  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:43.182654  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:43.182774  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:44.185316  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:44.185535  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:45.187380  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:45.187593  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:46.189215  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:46.189333  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:51.533411  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:51.533663  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:52.535520  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:52.535737  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:53.538100  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:53.538225  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
12/26-16:46:54.540459  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.1 -> 192.168.200.200
12/26-16:46:54.540576  [**] [1:100002:0] Someone pinging your network [**] [Priority: 0] {ICMP} 192.168.200.200 -> 192.168.200.1
```

Figure 4.15. An alert message after ICMP flood DOS attack.

For ARP Spoofing attack, ARPwatch was used to detect the attack. ARPwatch generate alerts when an abnormal use of the ARP protocol is detected. Look at the output that ARPwatch generates when changes are detected in ARP/IP assignments. And the ARPWatch output shows the changes in MAC and IP addresses. Fig 4.12 below shows a change in MAC and IP address in line one immediately when ARP spoofing process starts.



```
root@kali:~# sudo arpwatch -i eth0
root@kali:~# sudo tail -f /var/log/syslog
Apr 10 04:14:08 kali arpwatch: changed ethernet address 192.168.58.1 00:0c:29:c0
:bc:13 (00:50:56:c0:00:08) eth0
Apr 10 04:14:18 kali arpwatch: flip flop 192.168.58.2 00:50:56:e6:9a:28 (00:0c:2
9:5a:fc:1f) eth0
Apr 10 04:14:28 kali arpwatch: report: pausing (cdepth 3)
Apr 10 04:14:51 kali arpwatch: flip flop 192.168.58.242 00:0c:29:5a:fc:1f (00:0c
:29:c0:bc:13) eth0
Apr 10 04:15:01 kali arpwatch: flip flop 192.168.58.242 00:0c:29:c0:bc:13 (00:0c
:29:5a:fc:1f) eth0
Apr 10 04:15:11 kali arpwatch: flip flop 192.168.58.242 00:0c:29:5a:fc:1f (00:0c
:29:c0:bc:13) eth0
Apr 10 04:15:22 kali arpwatch: report: pausing (cdepth 3)
Apr 10 04:15:28 kali arpwatch: listening on eth0
Apr 10 04:15:38 kali arpwatch: changed ethernet address 192.168.58.242 00:0c:29:
c0:bc:13 (00:0c:29:5a:fc:1f) eth0
Apr 10 04:15:42 kali arpwatch: flip flop 192.168.58.242 00:0c:29:c0:bc:13 (00:0c
:29:5a:fc:1f) eth0
Apr 10 04:15:48 kali arpwatch: changed ethernet address 192.168.58.1 00:0c:29:c0
:bc:13 (00:50:56:c0:00:08) eth0
Apr 10 04:15:52 kali arpwatch: flip flop 192.168.58.242 00:0c:29:5a:fc:1f (00:0c
:29:c0:bc:13) eth0
Apr 10 04:15:58 kali arpwatch: flip flop 192.168.58.242 00:0c:29:5a:fc:1f (00:0c
```

Figure 4.16. A flip flop message

4.3. Prevention

For mitigation of attacks, iptables firewall was used. Iptables firewall is rule-based so rules were written. Using firewall system, defined firewall rules were set. And these rule sets were formulated basing on Source IP address, Destination IP, Source Port, Destination Port, Packet size and Protocol. Here the aspect of intelligence in this system is reflected by the use of decision making system which can decide whether to block or allow traffic depending on the rules set.

After snort analyzing packets sent via the network to check whether in the transmitted packets there are suspicious ones, if there is any attack an alert could be triggered. This could call for a defensive mechanism against the found attacks in the network.

By applying Iptables firewall, traffic was supervised in a way that whenever packets were sent to the network they were first filtered and analyzed whether they matched with the specified rule sets in the firewall and if they did they were allowed to go through the network and for those which never matched with the set rules were dropped. In this way only genuine packets were executed. With this mechanism, incoming and outgoing packets could be fully analyzed using set rules. Therefore, to prevent the earlier discussed attacks, iptables rules were defined as below;

To defend against SYN-flood attack, iptables script was written as follows;

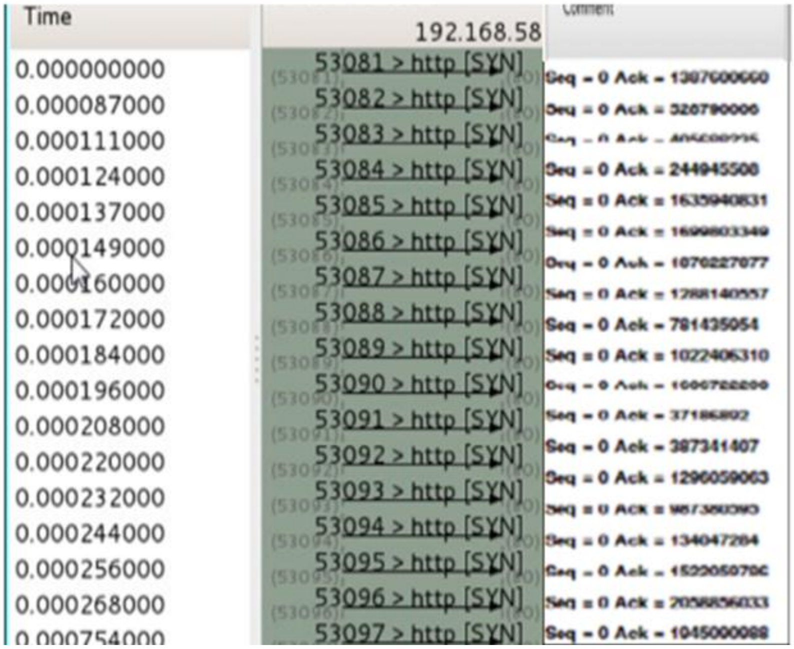
Table 4.1. Table showing iptable script against SYN-Flood attack.

```
# iptables -N syn_flood
#iptables -AINPUT-ptcp--syn-jsyn_flood
#  iptables  -A  syn_flood   -m  limit   --limit  1/s   --limit-burst  3   -j   RETURN
# iptables -A syn_flood -j DROP
```

---limit [rate[/unit]: the number of packets to let through per unit of time.

--limit-burst [count]: sets the count of packets that will be matched in a single “burst.

With this script, all incoming connections were allowed till the limit was reached. And it limited the number of incoming tcp connections or syn-flood attacks. First it will allow the normal working of the TCP connection establishment process. All network traffic going out of your machine will be allowed out, but all TCP/IP traffic coming into your machine will simply be dropped.



```
Time              192.168.58           Comment
0.000000000       53081 > http [SYN]   Seq = 0 Ack = 1307600660
0.000087000       53082 > http [SYN]   Seq = 0 Ack = 320790000
0.000111000       53083 > http [SYN]   Seq = 0 Ack = 405600296
0.000124000       53084 > http [SYN]   Seq = 0 Ack = 244945500
0.000137000       53085 > http [SYN]   Seq = 0 Ack = 1635940831
0.000149000       53086 > http [SYN]   Seq = 0 Ack = 1699803349
0.000160000       53087 > http [SYN]   Seq = 0 Ack = 1070227077
0.000172000       53088 > http [SYN]   Seq = 0 Ack = 1288140557
0.000184000       53089 > http [SYN]   Seq = 0 Ack = 781435054
0.000196000       53090 > http [SYN]   Seq = 0 Ack = 1022406310
0.000208000       53091 > http [SYN]   Seq = 0 Ack = 1006722200
0.000220000       53092 > http [SYN]   Seq = 0 Ack = 37186802
0.000232000       53093 > http [SYN]   Seq = 0 Ack = 387341407
0.000244000       53094 > http [SYN]   Seq = 0 Ack = 1296059063
0.000256000       53095 > http [SYN]   Seq = 0 Ack = 987380393
0.000268000       53096 > http [SYN]   Seq = 0 Ack = 134047284
0.000754000       53097 > http [SYN]   Seq = 0 Ack = 1522050706
                                       Seq = 0 Ack = 2058856033
                                       Seq = 0 Ack = 1045000088
```

Figure 4.17. shows TCP flow graph after iptables script applied to defend against syn-flood attack.

To defend against UDP Flood Attack, iptables script was written as below:

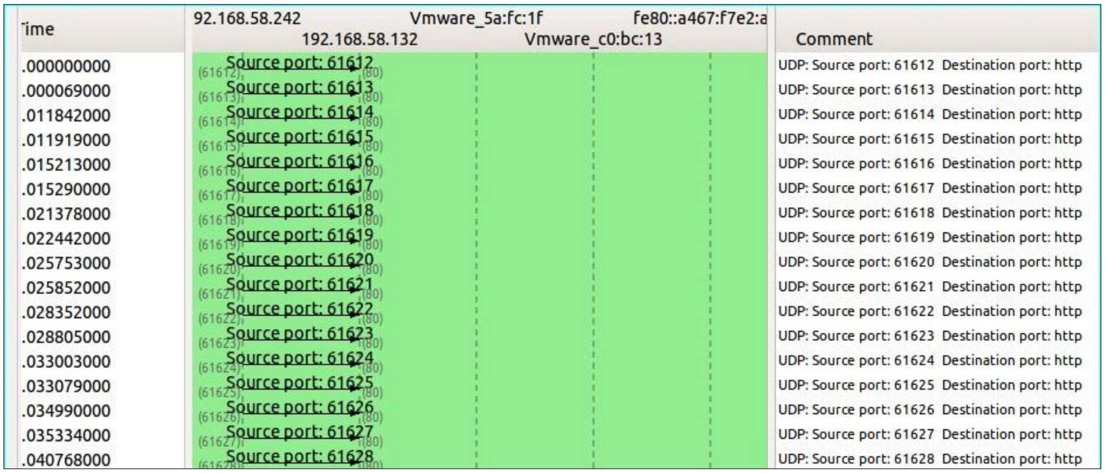
Table 4.2. Table showing Iptable script against UDP-Flood attack.

```
# iptables -N udp_flood
# iptables -A INPUT -p udp -j udp_flood
# iptables -A udp_flood -m state –state NEW –m recent –update –seconds 1 –hitcount 10 –j RETURN
# iptables -A udp_flood -j DROP
```

- **hitcount;** matches only if the hit-count for the packet's source address in the designated recent address list is at least *hits*.

- **NEW;** the packet is starting a new connection or is part of a connection that hasn't yet seen packets in both directions.

And this limits any IP address to 10 connections per minute and beyond this they are dropped. The script completely disables UDP packets continuously within 10 seconds of every connection.



```
ime          92.168.58.242        Vmware_5a:fc:1f       fe80::a467:f7e2:a
                   192.168.58.132        Vmware_c0:bc:13                       Comment
.000000000   (61612) Source port: 61612 (80)   UDP: Source port: 61612  Destination port: http
.000069000   (61613) Source port: 61613 (80)   UDP: Source port: 61613  Destination port: http
.011842000   (61614) Source port: 61614 (80)   UDP: Source port: 61614  Destination port: http
.011919000   (61615) Source port: 61615 (80)   UDP: Source port: 61615  Destination port: http
.015213000   (61616) Source port: 61616 (80)   UDP: Source port: 61616  Destination port: http
.015290000   (61617) Source port: 61617 (80)   UDP: Source port: 61617  Destination port: http
.021378000   (61618) Source port: 61618 (80)   UDP: Source port: 61618  Destination port: http
.022442000   (61619) Source port: 61619 (80)   UDP: Source port: 61619  Destination port: http
.025753000   (61620) Source port: 61620 (80)   UDP: Source port: 61620  Destination port: http
.025852000   (61621) Source port: 61621 (80)   UDP: Source port: 61621  Destination port: http
.028352000   (61622) Source port: 61622 (80)   UDP: Source port: 61622  Destination port: http
.028805000   (61623) Source port: 61623 (80)   UDP: Source port: 61623  Destination port: http
.033003000   (61624) Source port: 61624 (80)   UDP: Source port: 61624  Destination port: http
.033079000   (61625) Source port: 61625 (80)   UDP: Source port: 61625  Destination port: http
.034990000   (61626) Source port: 61626 (80)   UDP: Source port: 61626  Destination port: http
.035334000   (61627) Source port: 61627 (80)   UDP: Source port: 61627  Destination port: http
.040768000   (61628) Source port: 61628 (80)   UDP: Source port: 61628  Destination port: http
```

Figure 4.18 Shows flow graph after applied the iptables script against udp flood attack.

To defend against ICMP Flood Attack, iptables script was written:

Table 4.3. Iptable script against ICMP flood attack

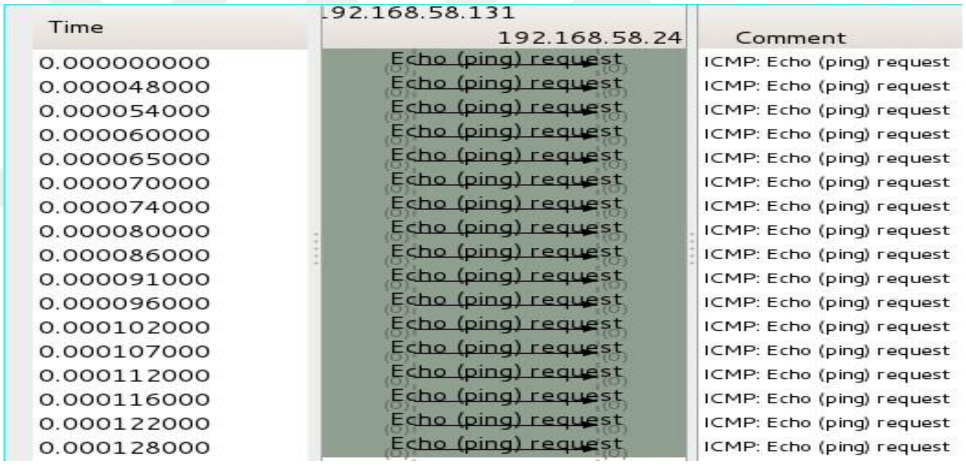
```
# iptables -N icmp_flood
# iptables -A INPUT -p icmp -j icmp_flood
#iptables -A icmp_flood -m limit --limit 1/s --limit-burst 3 -j RETURN
# iptables -A icmp_flood -j DROP
```

--limit [rate [/unit]]- the number of packets to let through per unit of time.

---limit-burst [count] - sets the count of packets that will be matched in a single "burst.

Inorder to limit the incoming icmp ping requests, the limit module was used to put a limit on the incoming connections. This rule could accept ping connections to 1 per second, with an initial burst of 3. An if this level crossed it would log the packet with a ping drop.

Then the next rule drops packets if it tries to cross this limit. This rule blocks all ICMP incoming packets from out side connection. Here we limit ICMP traffic to one packet per second. So even if someone floods the network via smurf, the they can not be allowed. Attacker is sending ICMP Echo packets continiously but victims machine is not responding by sending ICMP Reply packets as all the packets are being dropped by the firewall following Iptable rules.



Time	192.168.58.131 192.168.58.24	Comment
0.000000000	Echo (ping) request	ICMP: Echo (ping) request
0.000048000	Echo (ping) request	ICMP: Echo (ping) request
0.000054000	Echo (ping) request	ICMP: Echo (ping) request
0.000060000	Echo (ping) request	ICMP: Echo (ping) request
0.000065000	Echo (ping) request	ICMP: Echo (ping) request
0.000070000	Echo (ping) request	ICMP: Echo (ping) request
0.000074000	Echo (ping) request	ICMP: Echo (ping) request
0.000080000	Echo (ping) request	ICMP: Echo (ping) request
0.000086000	Echo (ping) request	ICMP: Echo (ping) request
0.000091000	Echo (ping) request	ICMP: Echo (ping) request
0.000096000	Echo (ping) request	ICMP: Echo (ping) request
0.000102000	Echo (ping) request	ICMP: Echo (ping) request
0.000107000	Echo (ping) request	ICMP: Echo (ping) request
0.000112000	Echo (ping) request	ICMP: Echo (ping) request
0.000116000	Echo (ping) request	ICMP: Echo (ping) request
0.000122000	Echo (ping) request	ICMP: Echo (ping) request
0.000128000	Echo (ping) request	ICMP: Echo (ping) request

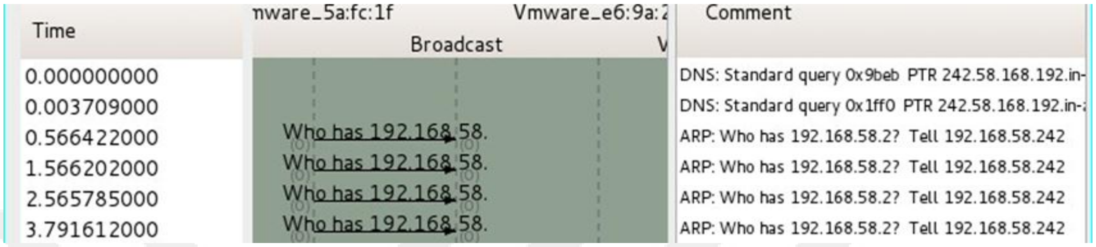
Figure 4.19. Shows flow graph after applied the iptable script against icmp flood attack.

To prevent spoofing of ip address the following script was used;

Table 4.4. Iptable script against arp spoofing

```
#iptables –A INPUT –I –s  -p tcp -m 192.168.58.242 –j LOG –log-prefix “192.168.58.2_spoof A“ ACCEPT
#iptables -A INPUT –eth1 –s 192.168.58.242  j DROP
```

This script logs and blocks IP spoofing on the ınterface called eth.1. And with this whenever an arp request is broadcast it could not give any reply (unicast) as shown in figure 4.20.



```
Time          nware_5a:fc:1f           Vmware_e6:9a:2   Comment
                        Broadcast                V
0.000000000                                             DNS: Standard query 0x9beb  PTR 242.58.168.192.in-
0.003709000                                             DNS: Standard query 0x1ff0  PTR 242.58.168.192.in-
0.566422000      Who has 192.168.58.                    ARP: Who has 192.168.58.2?  Tell 192.168.58.242
1.566202000      Who has 192.168.58.                    ARP: Who has 192.168.58.2?  Tell 192.168.58.242
2.565785000      Who has 192.168.58.                    ARP: Who has 192.168.58.2?  Tell 192.168.58.242
3.791612000      Who has 192.168.58.                    ARP: Who has 192.168.58.2?  Tell 192.168.58.242
```

Figure 4.20. shows TCP flow graph after iptables script applied to defend against arp spofing.

Furthermore, the time from when an attack was detected until firewall rules were applied was recorded for the test of the system. At most approximately 164 packets were sent as the system is detecting the DOS attack regarding the data given in Table 4.5. Prevention time is gained by dividing the maximum time the prevention process requires to the average time of single attack. 0.154 seconds of packet traffic (approximately 20 Mb data) is controlled by the attacker via ARP poisoning which is the difference between the maximum time the prevention process requires and the average time of a single attack. With this, it was easy to learn false alerts in the system.

Table 4.5. System Performance as Time Under 99% Confidence Level

Attack	Average Time of Single Attack	Average Prevention Time	Minimum Prevention Time	Maximum Prevention Time
DOS	0.00247	0.38888	0.37196	0.40573
ARP Poisoning	0.25807	0.39529	0.37858	0.41201

CHAPTER 5. CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

The objective of this thesis was to design an intelligent firewall agent as a defensive mechanism against network attacks. The intelligent firewall agent addresses two (2) important issues; that is to add intelligence to the traditional firewall and also to provide an effective defensive mechanism against network attacks. An intelligent firewall architecture was proposed to accomplish this objective. And it constituted four (4) components that is; the detection component, prevention component, end of attack component and lastly the cancel component.

The detection component was applied so as to detect any kind of intrusion in terms of attacks on the network. And for this case snort NIDS was used. Snort being rule based, rules were written basing on IP address, source IP, Destination IP address, and port numbers. This helped in way that whenever an attack would be done on the network, packets were checked whether they fit in the set snort rules, if packets matched the given set rule specifications they were allowed to go through the network and if no match was met packets were denied to go through the network. If probably an attack was detected from the IDS point, a prevention mechanism would be applied. The firewall rule based system was used. This was also built on rules. Firewall would monitor incoming and outgoing packets in the network.

Using this four way approach of detection, prevention, end of attack and cancel attacks, it proved fruitful than the other measures which were earlier implemented but did not seem to embrace all the four components of network defense mechanism at once. Thus, an intelligent firewall agent efficiently addressed the prevention of DOS attacks using iptable firewall which only allowed genuine packets to go through the network using set rules. In addition, intelligence was also successfully added to the existing firewall by using an expert system which improved on the accuracy and performance of the system by reducing false alerts.

5.2. Limitations

Despite achieving all the stated objectives by the study, there were some limitations incurred during the study; during the detection of ARP spoofing, a passive method was used to detect ARP spoofing. The main drawback of the passive method was a time lag between learning the address mappings and subsequent attack detection. In a situation where the ARP spoofing began before the detection tool was started for the first time, the tool could learn the forged replies in its IP to MAC address mapping database. And whenever the victim started communicating with some other hosts the inconsistency was also detected and alert was triggered to show that there is an attack yet in reality it was not an attack. So with this, false alerts were generated.

5.3. Recommendations

An automated firewall rule generating system is recommended for generating rules. Since there is a lot of increased human intervention during rule generation process and it needs to be reduced since this makes the whole process very prone to mistakes and it is also time consuming writing rule per rule.

REFERENCES

[1] Gupta, B., R.C. Joshi, and M. Misra, Distributed Denial of Service prevention techniques. arXiv preprint arXiv:1208.3557, 2012.

[2] Rodriguez, S. 60% of world's population still won't have Internet by the end of 2014. 2014 May/7/2014 [cited 2016 15/5/2016]; Available from: http://www.latimes.com/business/technology/la-fi-tn-60-world-population-3-billion-internet-2014-20140507-story.html.

[3] Brahima Sanou, The world in 2015-ICTFacts & Figures. 2015, International Telecommunication Union: Geneva,. p. 6.

[4] Kumaranayaka, D., et al., Intelligent Firewall Rule Generating System based on Passive Data Gathering. 2012.

[5] Rasmi, M. and A. Jantan, A New Algorithm to Estimate the Similarity between the Intentions of the Cyber Crimes for Network Forensics. Procedia technology, 2013. 11: p. 540-547.

[6] Anyoli, B.V.R.E. URA lost over sh2b in computer hacking scam. New Vision,Uganda 26th June 2012 [cited 2016 Wednesday,February 17,2016]; Available from: http://www.newvision.co.ug/new_vision/news/1302967/ura-lost-sh2b-hacking-scam.

[7] Thornton, G. Cyber attacks cost global business $300bn+. 2015 22 Sep 2015 [cited 2016 17/February/2016]; Available from: http://www.grantthornton.global/en/insights/articles/cyber-attacks-cost-global-business-over-$300bn-a-year/.

[8] Olivier, F., G. Carlos, and N. Florent, New Security Architecture for IoT Network. Procedia Computer Science, 2015. 52: p. 1028-1033.

[9] Stuart Mcclure, J.S.a.G.K., Hacking Exposed: Network Security Secrets & Solutions. 2005, McGraw-Hill/ Osborne.

[10] Lawrence C. Miller, C., Next-Generation Firewalls For Dummies. 2011 Indianapolis, Indiana: Wiley Pubishing,Inc. 76.

[11] Pranschke, G.-C., B. Irwin, and R.J. Barnett, Passive Traffic Inspection for Automated Firewall Rule Set Generation.

[12] Schultz, E.E., When firewalls fail: lessons learned from firewall testing. Network Security, 1997. 1997(2): p. 8-11.

[13] Hachana, S., N. Cuppens-Boulahia, and F. Cuppens, Mining a high level access control policy in a network with multiple firewalls. Journal of Information Security and Applications, 2015. 20: p. 61-73.

[14] Ioannidis, S., et al. Implementing a distributed firewall. in Proceedings of the 7th ACM conference on Computer and communications security. 2000. ACM.

[15] Roy, S., et al. A survey of game theory as applied to network security. in System Sciences (HICSS), 2010 43rd Hawaii International Conference on. 2010. IEEE.

[16] Alfantookh, A.A., DoS attacks intelligent detection using neural networks. Journal of King Saud University-Computer and Information Sciences, 2006. 18: p. 31-51.

[17] Melendez, W.A. and E.L. Petersen, The upper layers of the ISO/OSI reference model (Part II). Computer standards & interfaces, 1999. 20(4): p. 185-199.

[18] Les M Clellan, D.W., Sandy Workman, THE OSI REFERENCE MODEL. Cisco Systems.

[19] Stevens, W.R., TCP/IP Illustrated,Volume 1:The Protocols. first edition ed. Vol. 1. 1993: Addison Wesley. 600.

[20] Javvin, Network Protocols Handbook. 2nd Edition. ed. 2005: USA. 359.

[21] Meghanathan, N., A Tutorial on Network Security: Attacks and Controls. arXiv preprint arXiv:1412.6017, 2014.

[22] Fujitsu Network Communications, I. The TCP/IP Protocol Suite. 2006 December 20, 2006 [cited 2016 25/3/2016]; Available from: http://www.fujitsu.com/downloads/TEL/fnc/pdfservices/TCPIPTutorial.pdf.

[23] Parziale, L., et al., TCP/IP tutorial and technical overview. 2006: IBM Redbooks.

[24] Zimmermann, H., OSI Reference Model {The ISO Model of Architecture for Open Systems Interconnection. IEEE Transactions on Communications, 28 (4). 1980, April.

[25] Canavan, J.E., Fundamentals of Network Security. 2011, London.

[26] Fairhurst, G. Address Resolution Protocol (arp). 2015 [cited 2016 28/3/2016]; 1/12/2005:[Available from: http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.

[27] Cisco, Information About the Address Resolution Protocol. 2013: San Jose, USA.

[28] Systems, C., IP Addressing: ARP Configuration Guide. 2012.

[29] E.Comer, D., Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1. 1995, New Jersey: Alan Apt.

[30] Egli, P.R., INTRODUCTION TO RARP, BOOTP AND DHCP, PROTOCOLS FOR DYNAMIC DISTRIBUTION OF NETWORK CONFIGURATION PARAMETERS. 2015. p. 19.

[31] Schuba, C.L., et al. Analysis of a denial of service attack on TCP. in Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on. 1997. IEEE.

[32] Ousley, M.R., Information security the complete reference. 1993, San Fransisco.

[33] Ahmad, N. and M.K. Habib, Analysis of Network Security Threats and Vulnerabilities by Development & Implementation of a Security Network Monitoring Solution. 2010.

[34] Fenner, W.C., Internet group management protocol, version 2. 1997.

[35] Yang, G., Introduction to TCP/IP Network Attacks. Secure Systems Lab. November, 1997.

[36] Harris, B. and R. Hunt, TCP/IP security threats and attack methods. Computer Communications, 1999. 22(10): p. 885-897.

[37] Postel, J., Simple mail transfer protocol. Information Sciences, 1982.

[38] Daya, B., Network security: History, importance, and future. University of Florida Department of Electrical and Computer Engineering, 2013.

[39] Meenu Rani Dey, R.P., Renuka Bareth, History, Importance & Wonder of Network Security in Present. INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 2014. 1(ISSN: 2277-9655): p. 6.

[40] Tasevski, P. 2012 26/ 2012 [cited 2016 7/3/2016]; Available from: http://predragtasevski.com/posts/2012/10/what-is-network-security/.

[41] Murthy, B.R., V. Padmakar, and M.A. Vasavi, Significances and Issues of Network Security. International Journal of Advanced Research in Computer and Communication Engineering, 2014. Vol. 3(issue 6): p. 8.

[42] Uprit, A., Network Security Using Linux/Unix Firewall. International Journal of Scientific Engineering and Technology, 2014. Volume No.3(Issue No.3): p. 4.

[43] Pawar, M.V. and J. Anuradha, Network Security and Types of Attacks in Network. Procedia Computer Science, 2015. 48: p. 503-506.

[44] Chen, Y., et al., Detecting and Preventing IP-spoofed Distributed DoS Attacks. IJ Network Security, 2008. 7(1): p. 69-80.

[45] White, S. 2014 October 08 2014 [cited 2016 3/4/2016]; Available from: http://www.cgma.org/magazine/news/pages/201411089.aspx.

[46] McCullough, J., Beyond the Firewall Using a Layered Security Strategy to Address Internal Security Threats. SurfControl, Scotts Valley, California, 2003.

[47] Mogollon, M., Cryptography and Security Services: Mechanisms and Applications. 2007, Hershey • New York: Cybertech Publishing.

[48] Sharma, K., N. Khandelwal, and M. Prabhakar. An Overview of Security Problems in MANET. in ISEM International Conference, Bangkok, Thailand. 2011.

[49] Murthy, B.R., V. Padmakar, and M.A. Vasavi, Significances and Issues of Network Security.

[50] Chris Hare, K.S., Internet Firewalls and Network Security. Second Edition ed. 1996, Indianapolis: New Riders Publishing.

[51] Ahuja, H. and E.J. Gupta, Analysis of Malicious Data in Underwater Sensor Network. Analysis, 2012. 2(4): p. 967-971.

[52] Parmar, M.K. and H.B. Jethva, Survey on Mobile ADHOC Network and Security Attacks on Network Layer. International Journal of Advanced Research in Computer Science and Software Engineering, 2013. 3(11): p. 708-716.

[53] Zhang, H., J. Shi, and X. Chen, A Multi-Level Analysis Framework in Network Security Situation Awareness. Procedia Computer Science, 2013. 17: p. 530-536.

[54] Gahlot, D., et al., Network Security: it's time to take it seriously.

[55] Leidigh, B.C., Fundamental Principles of Network Security, A.P. Conversion, Editor. 2005.

[56] Stymantec. Security -Various types of network attacks. 2013 27 Dec 2013 [cited 26/4/2016 2016]; Available from: http://www.symantec.com/connect/articles/security-11-part-3-various-types-network-attacks.

[57] Inam Mohammad, R.P., Aashiya Khatoon, A Review of types of Security Attacks and Malicious Software in Network Security. International Journal of Advanced Research in Computer Science and Software Engineering, 2014. Volume 4(Issue 5, May 2014): p. 3.

[58] Rai, A.K., R.R. Tewari, and S.K. Upadhyay, Different types of attacks on integrated MANET-Internet communication. International Journal of Computer Science and Security, 2010. 4(3): p. 265-274.

[59] Haughn, M. passive attack. 2014 August 2014 [cited 2016 4/4/2016]; Available from: http://whatis.techtarget.com/definition/passive-attack.

[60] Dr. G. Padmavathi, M.D.S., A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks (IJCSIS) International Journal of Computer Science and Information Security, 2009. Vol. 4,: p. 9.

[61] Dr. Rajinder Singh, S.K., NETWORK SECURITY & VULNERABLE SECURITY ASPECTS. GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES, 2014: p. 5.

[62] Shrivastava, S. and S. Jain, A Brief Introduction of Different type of Security Attacks found in Mobile Ad-hoc Network. International Journal of Computer Science & Engineering Technology (IJCSET), 2013. 4(3).

[63] Roi Saltzman, A.S., Active Man in the Middle Attacks in A whitepaper from IBM Rational Application Security Group 2009, IBM Corporation.

[64] Jatinder Teji, R.C., Sonam mahajan, Manpreet Kaur Gill, Manju Dandi, Detection and Prevention of Passive Attacks in Network Security. International Journal of Engineering Science and Innovative Technology (IJESIT), 2013. Volume 2(Issue 6): p. 4.

[65] ComputerNetworkingNotes.com. ComputerNetworkingNotes.com. 2010 2010 [cited 2016 15/3/2016]; 2010:[Available from: http://computernetworkingnotes.com/network-security-access-lists-standards-and-extended/types-of-attack.html.

[66] Kumarasamy, S. and D.R. Asokan, An Efficient Detection Mechanism for Distributed Denial of Service (DDoS) Attack. arXiv preprint arXiv:1302.5158, 2013.

[67] Farraposo, S., L. Gallon, and P. Owezarski, eds. Network Security and DoS Attacks. 2005. 24.

[68] Salem, M. and H. Armstrong. Identifying DOS attacks using data pattern analysis. in Australian Information Security Management Conference. 2008.

[69] Russell, R., Hack Proofing Your Network. Second Edition ed. 2002, Rockland: Syngress Publishing, Inc.

[70] AL-Musawi, B.Q.M., Mitigating DoS/DDoS attacks using iptables. Int. J. Eng. Technol. IJET-IJENS, 2012. 12(03).

[71] Misra, S., et al., An adaptive learning routing protocol for the prevention of distributed denial of service attacks in wireless mesh networks. Computers & Mathematics with Applications, 2010. 60(2): p. 294-306.

[72] Saied, A., R.E. Overill, and T. Radzik, Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing, 2016. 172: p. 8.

[73] Hashmi, M.J., M. Saxena, and R. Saini, Classification of DDoS attacks and their defense techniques using intrusion prevention system. International Journal of Computer Science & Communication Networks, 2012. 2(5): p. 607-614.

[74] Trost, R., PRACTICAL INTRUSION ANALYSIS: Prevention and Detection for the Twenty-First Century. 2010, Crawfordsville, Indiana: Addison Wesley. 455.

[75] Arora, K., K. Kumar, and M. Sachdeva, Impact analysis of recent DDoS attacks. International Journal on Computer Science and Engineering, 2011. 3(2): p. 877-883.

[76] Douligeris, C. and A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art. Computer Networks, 2004. 44(5): p. 643-666.

[77] Bishop, M., Computer Security: Art and Science. 2002: Addison Wesley.

[78] Kumarasamy, S. and A. Gowrishankar, An Active Defense Mechanism for TCP SYN flooding attacks. arXiv preprint arXiv:1201.2103, 2012.

[79] Sanden, D.v.d., Detecting UDP attacks in high speed networks using packet symmetry with only flow data. 2008. p. 48.

[80] Chatterjee, K., Design and Development of a Framework to Mitigate DoS/DDoS Attacks Using IPtables Firewall. International Journal of Computer Science and Telecommunications, 2013. 4(3).

[81] Sieklik, B., R. Macfarlane, and W.J. Buchanan, Evaluation of TFTP DDoS amplification attack. Computers & Security, 2016. 57: p. 67-92.

[82] Suwala, P. and N. Wieczorek, Defense against DoS, flooding attacks.

[83] Symantec. Smurf DOS attack. 2015 [cited 2015 31/December]; Available from: https://www.symantec.com/security_response/glossary/define.jsp%3Fletter%3Ds%26word%3Dsmurf-dos-attack.

[84] Mobarhan, M.A., M.A. Mobarhan, and A. Shahbahrami, Evaluation of security attacks on UMTS authentication mechanism. International Journal of Network Security & Its Applications, 2012. 4(4): p. 37.

[85] Pateriya, P.K. and S.S. Kumar, Analysis on Man in the Middle Attack on SSL. International Journal of Computer Applications, 2012. 45(23).

[86] Net, M.T. Common Types of Network Attacks. 2016 2016 15/3/2016]; Available from: https://technet.microsoft.com/en-us/library/cc959354.aspx.

[87] Modi, C., et al., A survey of intrusion detection techniques in cloud. Journal of Network and Computer Applications, 2013. 36(1): p. 42-57.

[88] Verma, A. and A. Singh, An Approach to Detect Packets Using Packet Sniffing. International Journal of Computer Science and Engineering Survey, 2013. 4(3): p. 21.

[89] Shinde, P. and T.J. Parvat, DDoS Attack Analyzer: Using JPCAP and WinCap. Procedia Computer Science, 2016. 79: p. 781-784.

[90] Oluwabukola, O., et al., A Packet Sniffer (PSniffer) application for network security in Java. Issues in Informing Science and Information Technology, 2013. 10.

[91] Asrodia, P. and H. Patel, Network traffic analysis using packet sniffer. International Journal of Engineering Research and Applications, 2012. 2(3): p. 854-856.

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

[93] TechTarget. ping sweep (ICMP sweep). 2005 september/2005 [cited 2016 23/4/2016]; Available from: http://searchnetworking.techtarget.com/definition/ping-sweep-ICMP-sweep.

[94] Alhomoud, A., et al., Performance evaluation study of intrusion detection systems. Procedia Computer Science, 2011. 5: p. 173-180.

[95] Rehman, R.U., Intrusion Detection Systems with Snort, M. Sudul, Editor. 2003, Publishing as Prentice Hall PTR: New Jersey.

[96] Al-mamory, S.O. and F.S. Jassim, On the designing of two grains levels network intrusion detection system. Karbala International Journal of Modern Science, 2015. 1(1): p. 15-25.

[97] Folino, G. and P. Sabatino, Ensemble based Collaborative and Distributed Intrusion Detection Systems: A Survey. Journal of Network and Computer Applications, 2016.

[98] Raiyn, J., A survey of cyber attack detection strategies. International Journal of Security and Its Applications, 2014. 8(1): p. 247-256.

[99] Chirag Modi, D.P., Bhavesh Borisaniya, Hiren Patel, Avi Patel , Muttukrishnan Rajarajan A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications, 2012: p. 16.

[100] Richards, K., Network based intrusion detection: a review of technologies. Computers & Security, 1999. 18(8): p. 671-682.

[101] Sahithi Dandamudi, T.E., Firewalls Implementation in Computer Networks and their role in Network Security. 2015, University of Bridgeport. p. 6.

[102] Kaur, H. and M.A. Alm, Implementation of Portion Approach in Distributed Firewall Application for Network Security Framework. arXiv preprint arXiv:1201.4555, 2012.

[103] Broadband, A.N. Firewall Architecture: Understanding the purpose of a firewall when connecting to ADSL network services. 2011 [cited 2016 9/4/2016]; June 2011:[10]. Available from: http://www.tech2u.com.au/products/dsl/pdf/Firewall_Architecture.pdf.

[104] Fulp, D.E.W. Firewalls. 2009 [cited 2016 31/3/2016]; Available from: http://ac.els-cdn.com/B9780124166882000064/3-s2.0-B9780124166882000064-main.pdf?_tid=186741a8-f735-11e5-b59d-00000aacb362&acdnat=1459424492_8020470eab39f661549bc6cdb185b087.

[105] Ayuso, P.N., R.M. Gasca, and L. Lefevre, FT-FW: A cluster-based fault-tolerant architecture for stateful firewalls. computers & security, 2012. 31(4): p. 524-539.

[106] Trabelsi, Z., et al., Dynamic traffic awareness statistical model for firewall performance enhancement. computers & security, 2013. 39: p. 160-172.

[107] Leporati, A. and C. Ferretti, Modeling and Analysis of Firewalls by (Tissue-like) P Systems. SCIENCE AND TECHNOLOGY, 2010. 13(2): p. 169-180.

[108] Li, J., The research and application of multi-firewall technology in enterprise network security. Int'l J. of Security and Its Applications, 2015. 9(5): p. 153-162.

[109] Habtamu, A., An Overview of Firewall Technologies. Internet: heim. ifi. uio. no/~ abie/FirewallTechnologies. pdf, 2000.

[110] Bellovin, S.M. and W.R. Cheswick, Network firewalls. Communications Magazine, IEEE, 1994. 32(9): p. 50-57.

[111] Pundkar, M.S.G. and G. Bamnote, Analysis of Firewall Technology in Computer Network Security. 2014.

[112] Ko, H., Special Issues for Penetration testing of Firewall. 보안공학연구논문지 제권제호 년월 (Journal of Security Engineering), 2008. 5(4): p. 8.

[113] Miss. Shwetambari G. Pundkar1, P.D.G.R.B., Analysis of Firewall Technology in Computer Network Security. International Journal of Computer Science and Mobile Computing, 2014. Vol.3 (Issue.4): p. 6.

[114] Garcia-Alfaro, J., et al., Management of stateful firewall misconfiguration. Computers & Security, 2013. 39: p. 64-85.

[115] Asarcıklı, Ş., Firewall monitoring using intrusion detection systems. 2005.

[116] Scarfone, K. and P. Hoffman, Guidelines on Firewalls and Firewall Policy. 2009, National Institute of Standards and Technology: Gaithersburg,. p. 48.

[117] Yoo, I. and U. Ultes-Nitsche. An intelligent firewall to detect novel attacks–an integrated approach based on anomaly detection against virus attacks. in Proceedings of the SOFSEM conference, SOFSEM. 2002.

[118] Lanke, N.M. and C.R. Jacob, Detection of DDOS Attacks Using Snort Detection.

[119] Roesch, M. Snort: Lightweight Intrusion Detection for Networks. in LISA. 1999.

[120] Aickelin, U., J. Twycross, and T. Hesketh-Roberts, Rule Generalisation using Snort.

[121] Saxena, M. and A.K. Usman, Modelling Next Generation Intelligent Network Intrusion Prevention System using M-Key technique. International Journal of Computer Science, 2(01), 2013.

[122] Rehman, R.U., Advanced IDS Techniques Using Snort. Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. 2003, New Jersey.

[123] Purdy, G.N., Linux iptables Pocket Reference. First Edition. ed. 2004, Sebastopol: O’Reilly Media, Inc. 97.

[124] Linde, P., M. Pumputis, and G. Rodrıguez, iptables revisited: a not so ordinary ‘firewall’.

[125] Saive, R. Arpwatch Tool to Monitor Ethernet Activity in Linux. 2013 April 15, 2013.

RESUME

Hafuswa Nakato, I was born in Uganda on 24th.4.1988 in Mulago Hospital-Kampala District. I completed my Ordinary and Advanced levels of study from Bweyogerere secondary school. Later I joined Mbarara University of science and Technology in 2009 where I pursued Bachelors of Information Technology and I graduated in 2013. In Mbarara University of Science and Technology i was an active member of Mbarara University Computer Science Students Association.

Later in 2014 I joined Sakarya University, Department of Computer and Information Sciences Computer Engineering pursuing Masters in Computer and Information Engineering.