PACKET FILTERING USING IP TABLES IN LINUX

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science & Engineering

By:
Bhisham Sharma
(800832019)

Under the supervision of:
Mrs. Sanmeet Bhatia
Assistant Professor, SMCA

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

JUNE - 2010

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Packet Filtering using IP Tables in Linux"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mrs. *Sanmeet Bhatia* and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Bhisham

(Bhisham Sharma)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Sanmeet Bhatia

(Mrs. Sanmeet Bhatia)
Assistant Professor
SMCA
Thapar University
Patiala

Countersigned by

(Dr. Rajesh Bhatia) 12/07/10
Head,
Computer Science & Engineering Department,
Thapar University,
Patiala.

28/7/10

(Dr. R.K. Sharma)
Dean(Academic Affairs),
Thapar University,
Patiala.

Acknowledgement

I am highly thankful to my guide **Mrs. Sanmeet Bhatia**, Assistant Professor, SMCA, Thapar University, Patiala, for her advice, motivation, guidance, moral support, efforts and the attitude with which she solved all of my queries in making this thesis possible. It has been a great honor to work under him.

I am also thankful to **Dr. Rajesh Bhatia**, Head of Department, Computer Science and Engineering Department, Thapar University, Patiala, for providing us with adequate infrastructure in carrying the research work.

I am also thankful to **Ms. Inderveer channa**, P.G coordinator for providing us adequate environment, facility for carrying thesis work and all the staff member of the department for their full cooperation and help.

My greatest thanks are to all who wished me success especially my parents. Above all, I rendered my gratitude to the almighty who bestowed self-confidence, ability and strength in me to complete this work.

Bhisham

Bhisham Sharma
(800832019)

Abstract

The Internet is a fun little playground and at the same times a hostile environment. Like any other society, it's plagued with the kind of people who enjoy the electronic equivalent of writing on other people's walls with spray paint, tearing off their mailboxes, or just sitting in the street blowing their car horns. Some people get real work done over the Internet, and some must protect sensitive or proprietary data. Usually, a firewall's purpose is to keep the intruders out of the network while letting to do the job.

Nowadays, information is one of the very important assets in almost all organizations. Once the internal networks of those organizations are connected to the Internet, it becomes a potential target for cyber attacks. In order to secure the systems and information, each company or organization should conduct a self-hacking-audit, analyze the threats and eliminate it before getting any problem.

A firewall is a system or group of systems that enforces an access control policy between two or more networks. The means by which this control is accomplished varies widely, but in principle, the firewall is a pair of mechanisms, one that blocks traffic and one that permits traffic. Some firewalls emphasize blocking traffic, while others emphasize permitting traffic. The most important thing to recognize about a firewall is that it implements an access control policy.

In this thesis work major emphasis is on design and development of firewall script to deny/allow the network traffic. These scripts are written using the command line tool IP Tables, which support various features like the connection-tracking feature of IP Tables is a very useful thing. It can be used to prevent most TCP hijackings for non-IP Masqueraded clients that suffer from poor TCP sequence number randomization. Similarly, it can be used to prevent UDP packet hijacking in the same way.

Table of contents

List of Figures

List of Tables

Chapter1
Introduction

Computer networks by their very nature are designed to allow the flow of information. Network technology is such that, today, you can sit at a workstation in Delhi, and have a process connected to a system in London, with files mounted from a system in California, and be able to do work just as if all of the systems were in the same room. Impeding the free flow of data is contrary to the basic functionality of the network, but the free flow of information is contrary to the rules by which companies and governments need to conduct business. Information and sensitive data must be kept insulated from unauthorized access yet security must have a minimal impact on the overall usage of the network.

The purpose of a firewall is to provide a point of defense and a controlled and audited access to services, both from within and to an organizations private network. This requires a mechanism for selectively permitting or blocking traffic between the Internet and the network being protected. Routers can control traffic at an IP level, by selectively permitting or denying traffic based on source/destination address or port. Hosts can control traffic at an application level, forcing traffic to move out of the protocol layer for more detailed examination. To implement a firewall that relies on routing and screening, one must permit at least a degree of direct IP-level traffic between the Internet and the protected network.

1.1 Network Security

Network Security is a branch of Information Security which deals with systems that operate primarily at the network level. This includes the management of network devices such as Firewalls, VPNs, Proxies, NAC solutions, IDS/IPS, as well as the management and protection of the network infrastructure.

1.2 Network Security approaches

Security approaches are basically of following two types:

1.2.1 Proactive

Proactive approaches are measures that are taken to prevent computer or network from various types of attack. Every modern organization realizes the value of

dedicating some resources to the prevention of expensive damages that will likely to occur if such preventive measures are not taken. Banks use thick steel and concrete vaults with advanced electronic systems to prevent and detect break-ins. Some organizations have started using Intrusion Detection and Response Systems (IDRSes) to try to detect computer intrusions and then activate defensive measures when an attack is detected.

1.2.2 Reactive

Reactive approaches are those procedures that organizations use once they discover that some of their systems have been compromised by an intruder or attack program. Reactive methods include Disaster Recovery Plans, use of private investigation services and loss recovery specialists, reinstallation of operating systems and applications on compromised systems, or switching to alternate systems in other locations [1].

1.3 Network security objectives

Security objectives fall into one or more of the following categories:

1.3.1 Access Controls

Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. A system needs to be able to identify and authenticate users for access to data, applications and hardware. In a large system there may be a complex structure determining which users and applications have access to which objects.

1.3.2 Confidentiality

Confidentiality is the term used to prevent the disclosure of information to unauthorized individuals or systems. For example, a credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. Confidentiality is assurance that sensitive information remains private and is not visible to an eavesdropper. Confidentiality is critical to total data security. Encrypting data by using digital certificates and Secure Socket Layer (SSL) or virtual private network (VPN) connection helps ensure confidentiality when transmitting of data across the untrusted networks. Security policy should conclude how to provide confidentiality for information within network as well as when information leaves from network.

1.3.3 Availability

Availability is the prevention of unauthorized withholding of information. Information should be accessible and useable upon appropriate demand by an authorized user. Denial of service attacks are a common form of attack.

1.3.4 Integrity

Integrity is the unauthorized writing or modification of information. Integrity means that there is an external consistency in the system - everything is as it is expected to be. Data integrity means that the data stored on a computer is the same as the source documents. Data is protected from unauthorized changes or tampering. Data integrity defends against the security risk of manipulation, in which someone intercepts and changes information to which he or she is not authorized. When data enter into the system comes from a public network, need security methods to perform the following tasks:

- Protect the data from being sniffed and interpreted, typically by encrypting it.
- Ensure that the transmission has not been altered (data integrity).
- Prove that the transmission occurred (non-repudiation).

1.3.5 Non-repudiation

Non-repudiation is the prevention of either the sender or the receiver denying a transmitted message. A system must be able to prove that certain messages were sent and received. The use of digital certificates and public key cryptography to sign transactions, messages, and documents supports non-repudiation. Both the sender and the receiver agree that the exchange takes place. The digital signature on the data provides the necessary proof.

1.3.6 Authentication

The assurance or verification that the resource (human or machine) at the other end of the session really is what it claims to be. Solid authentication defends a system against the security risk of impersonation, in which a sender or receiver uses a false identity to access a system. When linking of system to a public network like the Internet, user authentication takes on new dimensions. Consequently, consider seriously the idea of using stronger authentication methods than traditional user name and password logon procedures provide. Authenticated users might have different types of permissions based on their authorization levels.

1.3.7 Authorization

Authorization is the assurance that the person or computer at the other end of the session has permission to carry out the request. Authorization is the process of determining who or what can access system resources or perform certain activities on a system. Typically, authorization is performed in context of authentication.

1.3.8 Auditing security activities

Auditing is basically monitoring of security-relevant events to provide a log of both successful and unsuccessful (denied) access. Successful access records tell who is doing what on your systems. Unsuccessful (denied) access records tell either that someone is attempting to break security or that someone is having difficulty accessing system [2].

1.4 Need of Network Security

Computer security technology is still in its infancy. Technologies such as firewalls, antivirus, and IDS have migrated from research labs into production networks, and have become required mainstays both as essential defenses and as legally mandated compliance systems. Computer security systems are complex devices that need to meet a variety of conflicting goals: high performance, fault tolerance, easy administration – and rigorous security processing. Some vendors have staked their claim based on speed, others on cost, and still others on the defensive posture and security of their products. Unfortunately, it is extremely difficult for the customer to sort through marketing fluff and dubious benchmarks, to determine which products actually work and which merely appear to work. Few customers are sufficiently sophisticated or willing to take the time to do their own testing and most are forced to rely on published results from trade magazines, recommendations from consultants, or industry analysts. Sadly, few of the trade magazines or analysts have the sophistication or time to perform adequate testing, either [3].

The network needs security against attackers and hackers. Network Security includes two basic securities. The first is the security of data information i.e. to protect the information from unauthorized access and loss. And the second is computer security i.e. to protect data and to thwart hackers. Here network security not only means security in a single network rather in any network or network of networks.

The need of network security has broken into two needs. One is the need of information security and other is the need of computer security.

On internet or any network of an organization, thousands of important information is exchanged daily. This information can be misused by attackers. The information security is needed for the following given reasons.

- To protect the secret information users on the net only. No other person should see or access it.
- To protect the information from unwanted editing, accidently or intentionally by unauthorized users.
- To protect the information from loss and make it to be delivered to its destination properly.
- To manage for acknowledgement of message received by any node in order to protect from denial by sender in specific situations. For example let a customer orders to purchase a few shares XYZ to the broker and denies for the order after two days as the rates go down.
- To restrict a user to send some message to another user with name of a third one. For example a user X for his own interest makes a message containing some favorable instructions and sends it to user Y in such a manner that Y accepts the message as coming from Z, the manager of the organization.
- To protect the message from unwanted delay in the transmission lines/route in order to deliver it to required destination in time, in case of urgency.
- To protect the data from wandering the data packets or information packets in the network for infinitely long time and thus increasing congestion in the line in case destination machine fails to capture it because of some internal faults.

Another part of network security includes the computer security. Computer security means to protect your computer system from unwanted damages caused due to network. One of the major reason for such damages are the viruses and spywares that can wipe off all the information from your hard disk or sometimes they may be enough destructive and may cause hardware problems too. Certainly the network must be protected from such type of damaging software's. The people who intentionally put such software on the network are called Hackers. As the network computers are part of it, so the computer security from Hackers is also a part of network security. The needs of computer security from Hackers are as follows:-

- It should be protected from replicating and capturing viruses from infected files.
- It needs a proper protection from virus and worms.
- There is a need of protection from Trojan Horses as they are enough dangerous for your computer [4].

IPtables is a packet filtering firewall used for blocking and allowing of network traffic on the basics of source address, destination address, port numbers, and protocols.

1.5 Packet Filtering using IP Tables

Packet filtering allows you to explicitly restrict or allow packets by machine, port, or machine and port. For instance, you can restrict all packets destined for port 80 (WWW) on all machines on your LAN except machine X and Y.

Packet filtering is most commonly used as a first line of defense against attacks from machines outside your LAN. Since most routing devices have built-in filtering capabilities, packet filtering has become a common and inexpensive method of security.

IPtables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Linux IPtables is currently the default firewall package that comes from RedHat, CentOS, UBUNTU and Fedora, right after ipchains dominated them long time ago. IPtables supports different types of filters. To name a few, IPtables can do filters and firewall rules by usernames, by group IDs and user profiles, by source and destination ports, by source host and destination hosts, by URLs, by IP addresses, by packet ID flags, by protocols, and a lot more including filtering by MAC address.

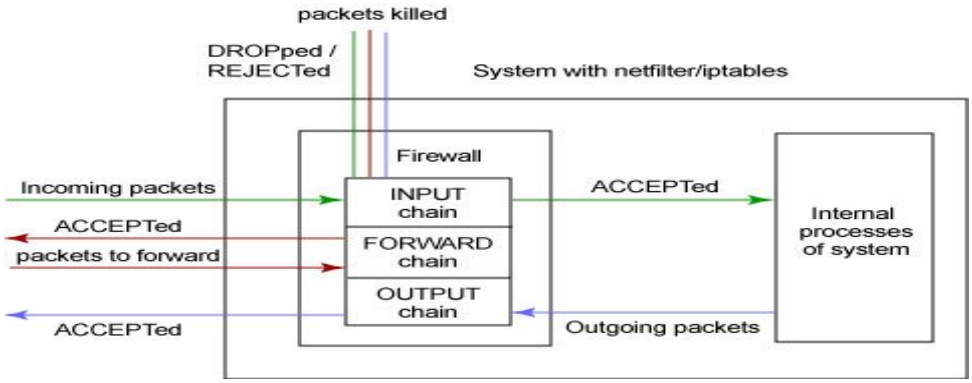


Figure1.1: Packet filtering process [5].

In figure 1.1 Firewall act as an intermediate between the incoming and outgoing traffic. Packets are accepted or dropped on the basics of rules applied in the INPUT, FORWARD, and OUTPUT chain.

The Linux kernel uses the IPtables facility to filter packets, allowing some of them to be received by or pass through the system while stopping others.

This facility is built in to the Linux kernel, and has three built-in tables or rules lists, as follows:

*filter—the default table for handling network packets.

*nat—used to alter packets that create a new connection and used for Network Address Translation (NAT).

*mangle—Used for specific types of packet alteration.

Each table has a group of built-in chains, which correspond to the actions performed on the packet by IPtables.

Every network packet received by or sent from a Linux system is subject to at least one table. However, a packet may be subjected to multiple rules within each table before emerging at the end of the chain. The structure and purpose of these rules may vary, but they usually seek to identify a packet coming from or going to a particular IP address, or set of addresses, when using a particular protocol and network service.

Regardless of their destination, when packets match a particular rule in one of the tables, a target or action is applied to them. If the rule specifies an ACCEPT target for a matching packet, the packet skips the rest of the rule checks and is allowed to continue to its destination. If a rule specifies a DROP target, that packet is refused access to the system and nothing is sent back to the host that sent the packet. If a rule specifies a QUEUE target, the packet is passed to user-space. If a rule specifies the optional REJECT target, the packet is dropped, but an error packet is sent to the packet's originator.

Every chain has a default policy to ACCEPT, DROP, REJECT, or QUEUE. If none of the rules in the chain apply to the packet, then the packet is dealt with in accordance with the default policy [6].

Chapter2

Literature Survey

2.1 Security Threats

Without security measures and controls in place, data might be subjected to an attack. Some attacks are passive, meaning information is monitored others are active, meaning the information is altered with intent to corrupt or destroy the data or the network itself. Networks and data are vulnerable to any of the following types of attacks if do not have a security plan in place.

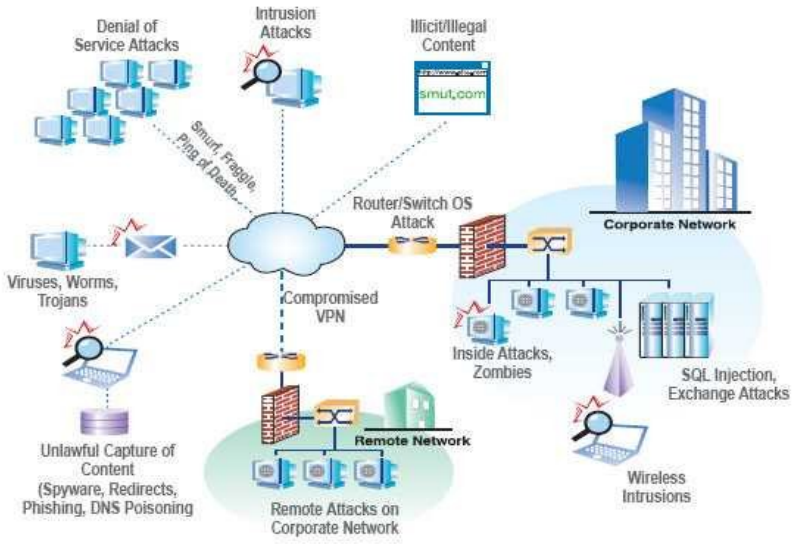


Figure 2.1: Major threats in today's network [7].

2.1.1 Denial of Service (DoS) Attacks

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root name servers. One common method of attack involves saturating the target (victim) machine with external communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. In general terms, DoS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or

obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

2.1.2 Website Defacement

Website defacement is an attack on a website that changes the visual appearance of the site. These are typically the work of system crackers, who break into a web server and replace the hosted website with one of their own. Website Defacement increasing tremendously experts no longer keep record of defaced sites. Attacker probes web services through normal Internet connection and modifies HTML or JAVA code, which changes website.

Website defacement is the unauthorized substitution of a web page or a part of it by a system cracker. This is a very common form of attack that seriously damages the trust and the reputation of a website. Detecting web page defacements is one of the main services for the security monitoring system.

2.1.3 Viruses and Worms

Viruses and Worms are computer programs that make computer systems not to work properly. There is a subtle difference between Virus and Worm; both can replicate itself, but when traveling on the network. Virus can't travel on its own on the network, where as Worms can travel on its own without anything. It doesn't actually need any infected file to stick in. Viruses and Worms are really annoying problem for all systems. The ultimate aim of these Viruses and Worms are making a good working system to malfunction and sometimes worms can sniff in and steal private information to send it to its creator. Earlier days, Viruses were spreading through floppy diskettes. Nowadays, it spreads through Internet, which is a broad gateway for these malicious programs. It can spread quickly and affect all systems in an organization within a minute and can create millions of dollar loss for the organization in a minute.

2.1.4 Data Sniffing and Spoofing

Data Sniffing and Spoofing attack are those in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

- **Sniffing**

  It means seeing all packets passed through wires or sometimes through air for wireless networks. Initially, this technique was being used for fixing network problems. Because it can watch network packets, it is now being used by hackers

for scanning login_ids and passwords over the wires. TCPdump and Wireshark are better examples for sniffing tools. The better way to avoid sniffing attack is encryption. If sensitive information is encrypted before sending to wires, hackers can't really understand what it is. They need the key to decrypt the information. This way, the information sent over network could always be safe with encryption. Typical services that are sniffed are: TELNET, FTP, SMTP (E-mail) packets if unencrypted.

- **Spoofing**

The exact meaning of spoofing is deceiving others. It is actually fooling other computer users to think that the source of their information is coming from a legitimate user. There are several methods of spoofing. Some of them are as follows:

- **IP Spoofing**

  It changes the source-address of an IP packet to show that it is from a legitimate source, but really it might be coming from a hacker. Thus, the hacker attacks the system and at the same time hides his IP address from the eyes of firewalls. The targeted systems for IP Spoofing are UNIX systems and RPC services.

- **DNS Spoofing**

  This will direct the users to incorrect location. In other words, directing the users to a different website and collecting personal information through web forms illegally.DNS Spoofing is actually very dangerous threat, because DNS is the one that manages domain names and creates equivalent IP addresses. Suppose, if the domain name is www.dell.com <http://www.dell.com/> and DNS calculates an IP address that is related to a hacker's site, the users will be directed to the hacker's website. If the hacker maintains his website similar to dell, then the users may think that the hacker's website is the real dell- website and may provide all bank or credit card information when trying to purchase something. Now, the hacker can get that information easily without any difficulties.

- **ARP Spoofing**

  ARP is actually maintaining a table of MAC addresses of all computers connected in a network. Any information that comes to ARP is delivered

to respective computer based on the mappings available on the ARP's tables. Suppose, if ARP couldn't find MAC address for a message, it broadcasts a message to all systems to get a reply from the exact destination-machine with its MAC address; when it gets the destination-machine's MAC address, it updates it on MAC table. This is the stage where ARP spoofing can happen. ARP Spoofing actually happens when a hacker (hacker's machine) sends a reply to the ARP's broadcasted message saying that the hacker's machine is the legitimate one. Then, ARP gets hacker's MAC address and adds it to its table. As a result, hacker will gain a legitimate connection to the network illegally. Once hacker is connected to the network, he can do all sorts of things.

2.1.5 Unauthorized Access

Unauthorized Access can be accomplished by any connection to a computer or network using most services (TELNET, FTP, HTTP, Web, E-mail, etc.). Hacker must somehow compromise authentication (password, token, PIN, Smart card) to gain access. Once access is gained malicious activity can occur unless internal auditing and access control is implemented, access can be undetected for years.

2.1.6 Man-in-the-Middle Attack

Man-in-the-middle attack occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data.

Man-in-the-middle attacks are like someone assuming identity in order to read message. The person on the other end might believe because the attacker might be actively replying to keep the exchange going and gain more information.

2.1.7 Trojan Horse

Hackers can use these programs to get control on their target machines and watch all the activities. This is very dangerous than Virus and DoS for the E-commerce businesses. The threatening issues with Trojan Horses are as follows:

- It allows for data integrity attack.
- It allows gaining control over the target machine and to steal private information available on the target system. This way it affects privacy policy.

- It can store key strokes and make it viewable for hackers. As a result, hackers can easily get the victim's login-ids and passwords. This way, it affects confidentiality.
- Hackers can see screen shots of targeted machines using Trojan horses. Sometimes, if websites are not secured properly, some third party companies can collect consumer information and pass it to some other businesses. It is a serious threat to customer privacy.
- It can be installed very easily on the target machines simply by sending it as an email attachment.

2.1.8 Port-scanning and Probing

Port-scanning and probing are techniques that identify vulnerable network ports:

- **Port-scanning**

  A port scanning is used to probe a network host for open ports. This is often used by administrators to verify security policies of their networks and by attackers to identify running services on a host with the view to compromising it. To port scan a host is to scan for listening ports on a single target host. To portsweep is to scan multiple hosts for a specific listening port. Port scanning is a technique that identifies vulnerable network ports or services (i.e. TELNET, FTP, E-mail, Web, etc) and its works by identifying as many targets as possible and tracking the ones those are receptive.

- **Probing**

  Once vulnerable ports are identified, the port can be probed with malicious intent [8, 9].

2.2 Security Measures

Network Security starts for authenticating any user. Once authenticated, firewall enforces access policies such as what services are allowed to be accessed by the network users. Though effective to prevent unauthorized access, this component fails to check potentially harmful contents such as computer worms being transmitted over the network. An Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) helps detect and prevent such malware.

2.2.1 Firewall

A firewall is a hardware or software solution to enforce security policies. In a physical security analogy, a firewall is equivalent to a door lock on a perimeter door or on a door to a room inside of the building – it permits only authorized user such as those with a key or access card to enter. A firewall has built-in filter that can disallow unauthorized or potentially dangerous material from entering the system. It also logs attempted intrusions.

2.2.2 Intrusion detection and prevention systems (IDPS)

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPS for other purposes, such as identifying problems with security policies, documenting existing threats, and determining individuals from violating security policies. IDPS have become a necessary addition to the security infrastructure of nearly every organization. IDPS typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPS can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content [10].

There are two main types of IDS's: network-based and host-based IDS.

- **Host based**

  The HIDS reside on a particular computer and provide protection for a specific computer system. Host intrusion detection systems are installed locally on host machines making it a very versatile system compared to NIDS. HIDS can be installed on many different types of machines namely servers, workstations and notebook computers. The model shown in Figure 2.2 allows for remote monitoring, remote storage of events logs and ability to PUSH agents to new or existing hosts [11].

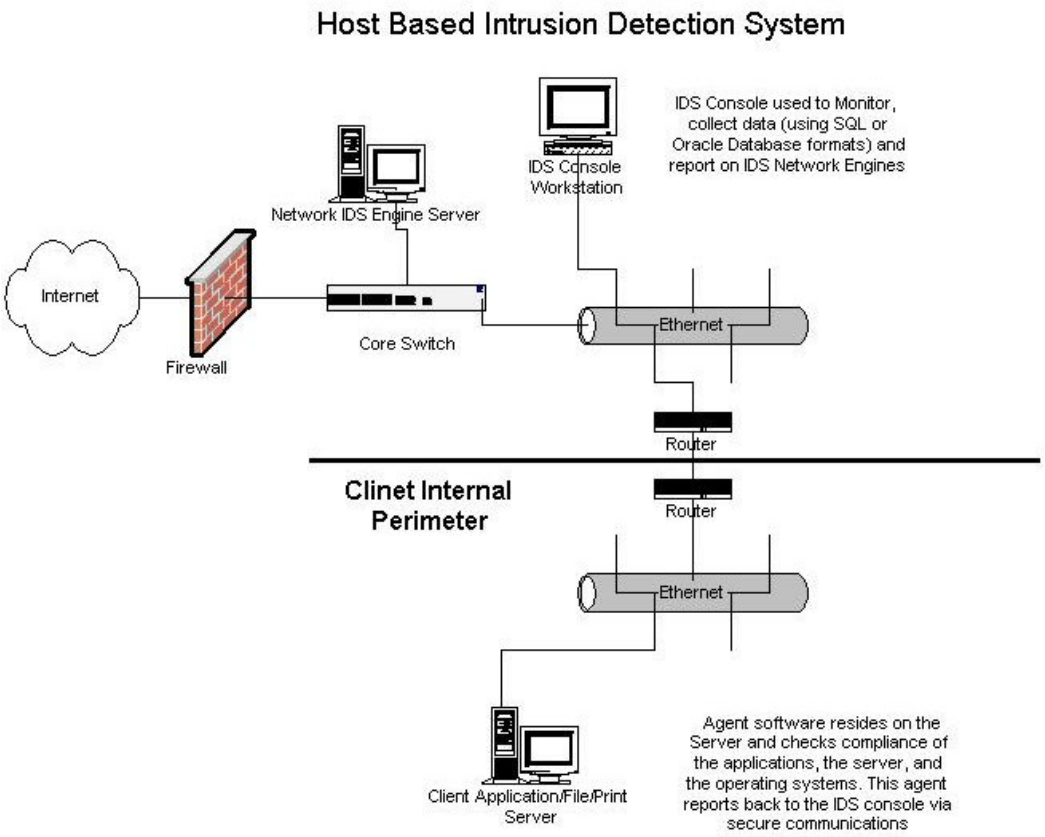


Figure 2.2: Host based Intrusion Detection System [12].

- **Network based**

  Network based IDS captures network traffic packets (TCP, UDP) and analyzes the content against a set of rules or signatures to determine if a possible event took place. NIDS monitors packets on the network wire and attempts to discover if a hacker/cracker is attempting to break into a system (or cause a denial of service attack). A typical example is a system that watches for large number of TCP connection requests (SYN) to many different ports on a target machine, thus discovering if someone is attempting a TCP port scan. A NIDS may run either on the target machine who watches its own traffic or on an independent machine promiscuously watching all network traffic (hub, router). NIDS is network based they do not only deal with packets going to a specific host – since all the machines in a network segment benefit from the protection of the NIDS. Network-based IDS can also be installed on active network elements, for example on routers. Typical Network Based IDS are Cisco Secure IDS, Hogwash, Dragon, and E-Trust IDS [13].

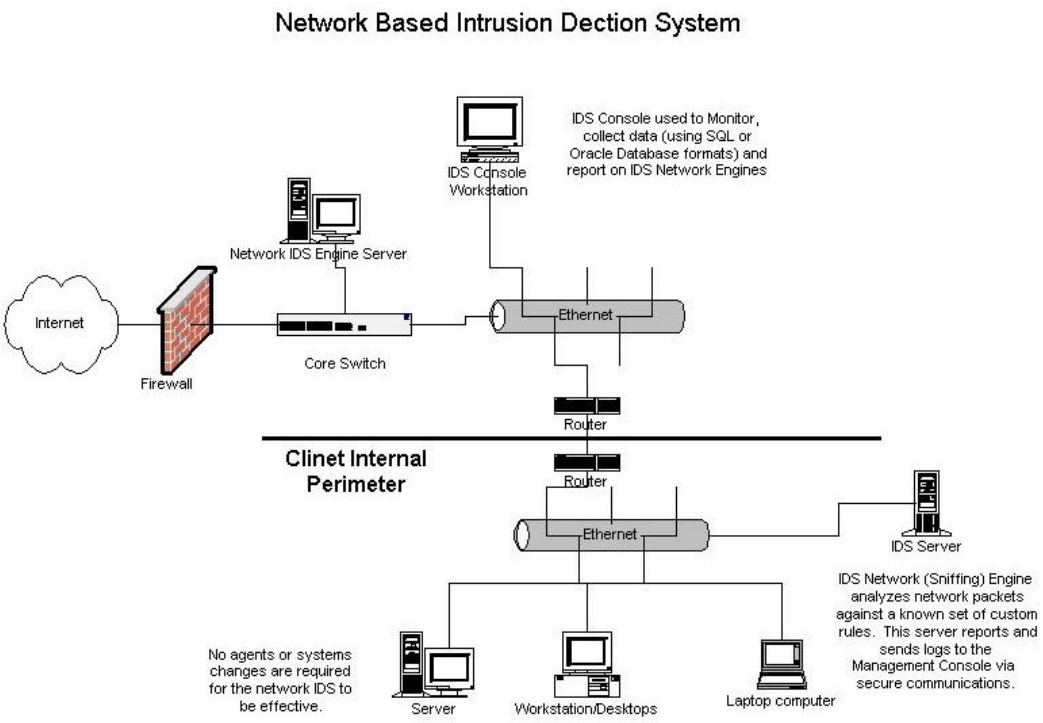


Figure 2.3: Network based Intrusion Detection System [12].

2.2.3 Virus Protection

Antivirus (or anti-virus) software is used to prevent, detect, and remove malware, including computer viruses, worms, and Trojan horses. Such programs may also prevent and remove adware, spyware, and other forms of malware.A variety of strategies are typically employed. Signature-based detection involves searching for known malicious patterns in executable code. However, it is possible for a user to be infected with new malware in which no signature exists yet. Some antivirus software can also predict what a file will do if opened/run by emulating it in a sandbox and analyzing what it does to see if it performs any malicious actions. If it does, this could mean the file is malicious.

2.2.4 Encryption

Encryption, which is the process of converting plaintext into some code called cipher text. Decryption is the reverse, in other words, moving from the cipher text back to plaintext. A cipher is a pair of algorithms which create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a key. This is a secret parameter ideally known only to the communicants. Keys are important, as ciphers without variable keys are trivially breakable and therefore less than useful for most purposes. Protects data in

transit or stored on disk. The act of ciphering and enciphering data through the use of shared software keys, data cannot be accessed without the appropriate software keys.

2.2.5 Data and Information Backups

Data and information backups must have for disaster recovery and business continuity. Should include daily and periodic (weekly) backups and be stored off-site, at least (20) miles away from geographic location, and have 24X7 access and be kept for at least (30) days while rotating stockpile.

Will mitigate the following attacks:

- Used to respond and replace information that is compromised by all the mentioned attacks [8, 9].

A key element in the protection of a computer connected to the Internet is the firewall. A firewall is like the door of our house; it sets a border between our private space and public space and allows us to decide who may enter and who can not. If our house had no door, any person could enter and search it. The same is true with our computer (the house) and firewall (the door). Without a firewall, anyone can enter the computer and see the files stored there, which may contain sensitive information and/or stuff as access codes and users names.

2.3 Firewall

A firewall is a logical object (hardware and/or software) within a network infrastructure which prevents communications forbidden by the security policy of an organization from taking place, analogous to the function of firewalls in building construction. Often a firewall is also referred to as a packet filter. The basic task of a firewall is to control traffic between different zones of trust and/or administrative authorities. Typical zones of trust include the Internet (a zone with no trust) and an internal network (a zone with high trust). The ultimate goal is to provide controlled connectivity between zones of differing trust levels through the enforcement of a security policy and a connectivity model based on the least privilege principle. Proper configuration of firewalls demands skill from the administrator. It requires considerable understanding of network protocols and of computer security. Small mistakes can lead to a firewall configuration worthless as a security tool and, in extreme situations, fake security where no security at all is left [14, 16].

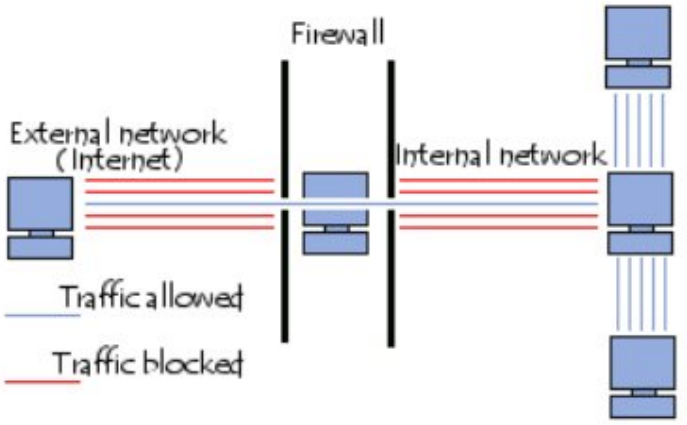


Figure 2.4: Firewall System [14].

In figure 2.4: Firewall that can protect internal network from the external network by accept/deny the traffic according to the rules specified in the list. A firewall is a system that protects a computer or a computer network against intrusions coming from a third-party network (generally the Internet). A firewall is a system that filters data packets that are exchanged over the network. Therefore, it is a filtering gateway that comprises at least the following network interfaces:

- An interface for the network being protected (internal network)
- An interface for the external network

2.3.1 There are a number of components that make up a firewall

- The Internet access security policy of the organization. This state, at a high Level, what degree of security the organization expects when connecting to the Internet. The security policy is independent of technology and techniques, and should have a lifetime independent of the equipment used. An example of statements from such a security policy might be: external users will not be allowed to access the corporate network without a strong level of authentication, any corporate information not in the public domain must be transferred across the Internet in a confidential manner, and corporate users will only be allowed to send electronic mail to the Internet - all other services will be banned.
- The mapping of the security policy onto technical designs and procedures that are to be followed when connecting to the Internet. This information will be updated as new technology is announced, and as system configurations change etc. For example, regarding authentication, the technical design might specify

the use of one-time passwords. Technical designs are usually based on one of two security policies, either:

- Permit any service unless it is expressly denied, or
- Deny any service unless it is expressly permitted.

The latter is clearly the more secure of the two.

- The firewall system, which is the hardware and software which implements the firewall. Typical firewall systems comprise an IP packet filtering router, and host computer (sometimes called a bastion host or application gateway) running application filtering and authentication software [14].

2.3.2 Advantages of Firewalls

Firewalls have a number of advantages.

- They can stop incoming requests to inherently insecure services, e.g. you can disallow rlogin, or RPC services such as NFS.
- They can control access to other services e.g. bar callers from certain IP addresses, filter the service operations (both incoming and outgoing), e.g. stop FTP writes, hide information e.g. by only allowing access to certain directories or systems.
- They are more cost effective than securing each host on the corporate network since there are often only one or a few firewall systems to concentrate on.

2.3.3 Disadvantages of Firewalls

Firewalls are not the be all and end all of network security. They do have some disadvantages, such as:

- They are a central point for attack, and if an intruder breaks through the firewall they may have unlimited access to the corporate network.
- They may restrict legitimate users from accessing valuable services, for example, corporate users may not be let out onto the Web, or when working away from home a corporate user may not have full access to the organization's network.
- They can be a bottleneck to throughput, since all connections must go via the firewall system.
- The biggest disadvantage of a firewall is that it gives no protection against the inside attacker. Since most corporate computer crime is perpetrated by internal

users, a firewall offers little protection against this threat. E.g. an employee may not be able to Email sensitive data from the site, but they may be able to copy it onto a floppy disc and post it [15].

2.3.4 Firewalls, Layers and Models

The working of firewall in different layers and models are shown below as:

Table 2.1: Firewalls Layers and Models [14].

ISO 7 Layer Model	Internet 5 Layer Model	Firewalls
Application (7)	Application (5)	Proxy Service
Transport (4) Network (3)	TCP/UDP (4) IP/ICMP (3)	Packet Filtering Router/ Packet Screening Router Stateful Inspection
Link (2) Physical (1)	Link (2) System Interface (1)	None

In table 2.1 ISO uses a 7 layer model for Open Systems Interconnection, whereas the Internet can be regarded as having a 5 layer model. Firewall systems are usually placed at layers 3, 4 and 5 of the Internet model, (3, 4 and 7 of the ISO model). Their purpose is to control access to and from a protected network. Firewall can be placed between any two networks, for example between a corporate business network and its R&D network. In general, a firewall is placed between a high security domain and a lower security domain. A firewall system operating at layers 3 and 4 is sometimes called a packet filtering router or a screening router. Its purpose is to filter IP and ICMP packets and TCP/UDP ports. The router will have several ports and be able to route and filter the packets according to the filtering rules. Packet filters can also be built in software and run on dual homed PCs, but whilst these can filter packets they are not able to route them to different networks. A firewall at layer 5 Internet (7 ISO) is sometimes called a bastion host, application gateway, proxy server or guardian system. Its purpose is to filter the service provided by the application [14].

2.3.5 Types of Firewalls

Firewalls are classified into three basic types:

2.3.5.1 Stateless Packet Filtering

A firewall system operates on the principle of simple packet filtering, or stateless packet filtering. It analyses the header of each data packet (datagram) exchanged

between an internal network computer and an external computer. Thus, the data packets exchanged between an external network computer and an internal network computer pass through the firewall and contain the following headers, which are systematically analyzed by the firewall:

- The IP address of the computer sending the packets
- The IP address of the computer receiving the packets
- The type of packet (TCP, UDP, etc.)
- The port number (port is a number associated with a service or a network application).

The IP addresses contained in the packets allow you to identify the computer that is sending the packets and the target computer, while the type of packet and the port number indicate the type of service being used.

The table below gives examples of firewall rules

Table 2.2: Firewall rules [16].

Rule	Action	Source IP	Target IP	Protocol	Source Port	Target Port
1	Accept	192.168.10.20	194.154.192.3	TCP	Any	25
2	Accept	Any	192.168.10.3	TCP	Any	80
3	Accept	192.168.10.0/24	Any	TCP	Any	80
4	Deny	Any	Any	Any	Any	Any

Table 2.2 describes four firewall rules for allowing and denying the network packet based on Source IP, Target IP, Protocol, Source Port, and Target Port. In first three rules action is accept because defining either the Source IP or Target IP to indicate that the packet is originating from the authorized network. But in fourth rule action is deny because Source IP, Target IP, Source Port, and Target Port are unidentified.

Recognized ports (whose numbers are between 0 and 1023) are associated with ordinary services (e.g. ports 25 and 110 are associated with e-mail and port 80 with the Web). Most firewall devices are at least configured to filter communications according to the port being used. It is generally recommended to block all ports that are not essential (depending on the security policy in place). For example, port 23 is often blocked by default by firewall devices because it corresponds to the TELNET protocol, which allows a person to emulate terminal access to a remote machine in

order to remotely execute commands. The data exchanged over TELNET are not encrypted, which means that a hacker is likely to listen to the network and steal any unencrypted passwords. Administrators generally prefer the SSH protocol, which has a reputation for being safe and provides the same functionalities as TELNET.

- **Stateless Packet Filtering - Pros**

1. They are fast because they operate on IP addresses and TCP/UDP port numbers alone, ignoring the data contents (payload) of packets.
2. Due to the fact that packet payload is ignored, application independence exists.
3. Least expensive of the three types of firewalls.
4. Packet filtering rules are relatively easy to configure.

- **Stateless Packet Filtering – Cons**

1. Allow a direct connection between endpoints through the firewall. This leaves the potential for a vulnerability to be exploited.
2. There is no screening of packet payload available. It is impossible to block users from visiting web sites deemed off limits, for example. Logging of network traffic includes only IP addresses and TCP/UDP port numbers, no packet payload information is available.
3. Complex firewall policies are difficult to implement using filtering rules alone.
4. There is a reliance on the IP address for authentication rather than user authentication [16].

2.3.5.2 Dynamic Filtering

Stateless Packet Filtering only attempts to examine the IP packets independently, which corresponds to level 3 of the OSI model. But most connections are supported by TCP protocol, which manages sessions, in order to ensure that exchanges take place smoothly. In addition, many services (e.g. FTP) initiate a connection on a static port but dynamically (i.e. randomly) open a port in order to establish a session between the machine acting as a server and the client machine. Thus, with stateless packet filtering it is impossible to anticipate which ports should be authorized and which should be prohibited. To remedy this situation, the system of dynamic packet filtering is based on the inspection of layers 3 and 4 of the OSI model, allowing for full monitoring of the transactions between the client and the server. The term for this

is "stateful inspection" or "stateful packet filtering". A "stateful inspection" firewall device is able to ensure the monitoring of exchanges, meaning that it takes into account the status of previous packets when defining filtering rules.

- **Dynamic Filtering-Pros**

1. Offers improved security over basic packet filters due to packet examination.
2. Offers a degree of application independence, based on level of stateful packet examination.
3. Better logging of activities over basic packet filters.
4. Good performance.

- **Dynamic Filtering-Cons**

1. Allow a direct connection between endpoints through the firewall. This leaves the potential for a vulnerability to be exploited.
2. No hiding of your private systems.
3. Setting up stateful packet examination rules is more complicated [17, 18].

2.3.5.3 Application proxies

Application filtering allows you to filter communications application by application. Application filtering operates at level 7 (the application layer) of the OSI model, contrary to simple packet filtering (level 4). Application filtering implies knowledge of the protocols used by each application. Application filtering implies knowledge of the applications on the network and notably of the way in which it structures the exchanged data (ports, etc.). A firewall performing application filtering is generally called an "application gateway" (or a "proxy") because it acts as a relay between two networks by intervening and performing a thorough evaluation of the content in the exchanged packets. Therefore, the proxy represents an intermediary between the internal network's computers and the external network, putting the attacks in their place. Moreover, application filtering allows for the headers that precede application messages to be destroyed, which provides an additional level of security. This type of firewall is highly effective and ensures good network protection if it is correctly run. On the other hand, detailed analysis of application data requires a lot of computing power, which often means slowed communications because each packet must be thoroughly analyzed.

- **Application proxies-Pros**

1. Has the best content filtering capability.

2. Can hide private systems.
3. Robust user authentication.
4. Firewall does not let end points communicate directly with one another. Thus vulnerability in a protocol which could slip by a packet filter or stateful packet inspection firewall could be overcome by the proxy program.
5. Offers the best logging of activities.
6. Policy rules are usually easier than packet filtering rules.

- **Application proxies-Cons**

1. Performance problems; much slower than the other two.
2. Must have a proxy for every protocol. Failure to have a proxy may prevent a protocol from being handled correctly by the firewall.
3. TCP is the preferred transport. UDP may not be supported.
4. No protection from all protocol weaknesses [19].

2.4 Firewall using IP Tables

IPtables provides comparatively higher speed and reliability than the other firewall tools. Being a Linux product, its integration with the OS is also more robust and reliable. It keeps a stateful track of each connection passing through it and tries to anticipate future actions. Its capacity of filtering packets on the basis of MAC address makes it a formidable security system. It can filter out attacks using malformed packets and can restrict access from locally attached servers to other networks in spite of their valid IP addresses. Network address translation (NAT) with masquerading capability of IPtables helps it to hide internal network IP sub networks behind one or a small poll of external IP addresses. NAT and masquerading enables the firewall system to open and close ports on the gateway. NAT and masquerading is very much helpful for network traffic analysis and devising relevant security measures. The rate-limiting feature of IPtables can block attacks even from some types of DoS (denial of service) attacks [20].

2.4.1 Linux IP Tables

IPtables is a current Linux Firewall mechanism and a successor of ipfilter and ipchains.The primary purpose is packet filtering based on header fields, e.g., IP addresses, TCP and UDP ports, and TCP flags. Originally, the most popular firewall/NAT package running on Linux was ipchains, but it had a number of

shortcomings. To rectify this, the Netfilter organization decided to create a new product called IPtables, giving it such improvements as:

- Better integration with the Linux kernel with the capability of loading IPtables-specific kernel modules designed for improved speed and reliability.
- Stateful packet inspection. This means that the firewall keeps track of each connection passing through it and in certain cases will view the contents of data flows in an attempt to anticipate the next action of certain protocols. This is an important feature in the support of active FTP and DNS, as well as many other network services.
- Filtering packets based on a MAC address and the values of the flags in the TCP header. This is helpful in preventing attacks using malformed packets and in restricting access from locally attached servers to other networks in spite of their IP addresses.
- System logging that provides the option of adjusting the level of detail of the reporting.
- Better network address translation [20].

How to start IP Tables

IPtables can start, stop, and restart after booting by using the commands:

[root@bhisham /]# service iptables start

[root@bhisham /]# service iptables stop

[root@bhisham /]# service iptables restart

To get IPtables configured to start at boot, use the chkconfig command:

[root@bhisham /]# chkconfig iptables on

2.4.2 Linux Packet filtering types

Four types of Linux Packet Filtering are classified as follows:

- **Ipfw**

  The first type of Linux packet filtering support with Linux 1.2 kernels. It is Stateless and very limited services like only filtered on one port, never integrated into distributions.

- **Ipfwadm**

  The second type of Linux packet filtering support with Linux 2.0 kernels. It is also Stateless and filters on source and destination addresses and ports and support for TCP, UDP, and ICMP protocols.

- **Ipchains**

  The third type of Linux packet filtering support with Linux 2.2 kernels. It is also Stateless and support for ICMP subtypes protocols other than TCP, UDP and ICMP, and inverse options.

- **IP Tables**

  The fourth type of Linux packet filtering support with Linux 2.4, 2.5, and 2.6 kernels. It is Stateful and support for both IPV4 and IPV6 and provides backward compatibility modules for ipfwadm and ipchains [21, 22].

2.4.3 Basic uses of IP Tables

The most common use of IPtables is to simply block and allow traffic.

Example1: To allow inbound "tcp" traffic to port 139

#iptables -A INPUT -p tcp --dport 139 -j ACCEPT

-A - this tells IPtables to "append" the new rule to the current IPtables rule set.

INPUT - The new rule will be appended to the "INPUT" portion of the rule set, which controls inbound server traffic.

-p - Indicates what protocol the rule applies to. Popular protocols are "tcp", "udp", "icmp" and several others.

--dport - Specifies the destination port to which the traffic will be directed. In this case, it's port 139.

-j - Instructs the firewall to "jump" to specified state. In this case, request to TCP port 139 "jump" to "ACCEPT" and are therefore accepted and allowed to pass through the firewall.

ACCEPT - Any inbound traffic to TCP port 139 will "jump" to an "ACCEPT" state, and thus will be able to pass through the firewall.

Example2: To allow incoming traffic on port 22 (traditionally used by SSH).

 # iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT

Specifically, this appends (-A) to the table INPUT the rule that any traffic to the interface (-i) eth0 on the destination port for Ssh that IPtables should jump (-j), or perform the action, ACCEPT [22, 23].

2.4.4 Processing for packets routed by the firewall

All packets inspected by IPtables pass through a sequence of built-in tables (queues) for processing. Each of these queues is dedicated to a particular type of packet activity and is controlled by an associated packet transformation/filtering chain.

Table 2.3: Packet Processing [20].

Queue Type	**Queue Function**	**Packet Transformation Chain in Queue**	**Chain Function**
Filter	Packet Filtering	Forward	Filters packets to servers accessible by another NIC on the firewall.
		Input	Filters packets destined to the firewall.
		Output	Filters packets originating from the firewall
Nat	Network Address Translation	Pre-routing	Address translation occurs before routing. Facilitates the transformation of the destination IP address to be compatible with the firewall's routing table. Used with NAT of the destination IP address, also known as destination NAT or DNAT.
		Post-routing	Address translation occurs after routing. This implies that there was no need to modify the destination IP address of the packet as in pre-routing. Used with NAT of the source IP address using either one-to-one or many-to-one NAT. This is known as source NAT, or SNAT.
		Output	Networks address translation for packets generated by the firewall.
Mangle	TCP header modification	Pre-routing Post-routing Output Input Forward	Modification of the TCP packet quality of service bits before routing occurs. (Rarely used in SOHO environments)

In table 2.3 there are three tables in total. The first table is the **filter** queue which is responsible for packet filtering. It has three built-in chains in which firewall policy rules are placed. These are the:

Forward chain: Filters packets to servers protected by the firewall.

Input chain: Filters packets destined for the firewall.

Output chain: Filters packets originating from the firewall.

The second table is the **Nat** queue which is responsible for network address translation. It has two built-in chains; these are:

Pre-routing chain: NATs packets when the destination address of the packet needs to be changed.

Post-routing chain: NATs packets when the source address of the packet needs to be changed.

The third is the **mangle** table which is responsible for the alteration of quality of service bits in the TCP header.

It is necessary to specify the table and the chain for each firewall rule you create. There is an exception: Most rules are related to filtering, so IPtables assumes that any chain that's defined without an associated table will be a part of the filter table. The filter table is therefore the default.

2.4.5 Targets and jumps in IP Tables

Each firewall rule inspects each IP packet and then tries to identify it as the target of some sort of operation. Once a target is identified, the packet needs to jump over to it for further processing.

Descriptions of the most commonly used targets

Table 2.4: Target and Jumps [20].

Target	**Description**
ACCEPT	IPtables stops further processing. The packet is handed over to the end application or the operating system for processing.
DROP	IPtables stops further processing. The packet is blocked.
LOG	The packet information is sent to the syslog daemon for logging. IPtables continues processing with the next rule in the table. As you can't log and drop at the same time, it is common to have

	two similar rules in sequence. The first will log the packet, the second will drop it.
REJECT	Works like the DROP target, but will also return an error message to the host sending the packet that the packet was blocked.
DNAT	Used to do destination network address translation. .i.e. rewriting the destination IP address of the packet.
SNAT	Used to do source network address translation rewriting the source IP address of the packet. The source IP address is user defined.
MASQUERADE	Used to do Source Network Address Translation. By default the source IP address is the same as that used by the firewall's interface.

2.4.6 Options

The options that are recognized by IPtables can be divided into several different groups.

2.4.6.1 COMMANDS

These options specify the specific action to perform.

-A --append *chain rule-specification*

Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

-D --delete *chain rulenum*

Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

-I --insert *chain* [*rulenum*] *rule-specification*

Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified.

-R --replace *chain rulenum rule-specification*

Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

-L --list [*chain*]

List all rules in the selected chain. If no chain is selected, all chains are listed. As every other IPtables command, it applies to the specified table (filter is the default), so NAT rules get listed by: iptables -t nat -n –L.

-F --flush [*chain*]

Flush the selected chain (all the chains in the table if none is given). This is equivalent to deleting all the rules one by one.

-Z --zero [*chain*]

Zero the packet and byte counters in all chains.

-N --new-chain *chain*

Create a new user-defined chain by the given name. There must be no target of that name already.

-X --delete-chain [*chain*]

Delete the optional user-defined chain specified. There must be no references to the chain. If there are, you must delete or replace the referring rules before the chain can be deleted. The chain must be empty, i.e. not contain any rules. If no argument is given, it will attempt to delete every non-built in chain in the table.

-P --policy *chain target*

Set the policy for the chain to the given target. Only built-in (non-user-defined) chains have policies, and neither built-in nor user-defined chains can be policy targets.

-E --rename-chain *old-chain new-chain*

Rename the user specified chain to the user supplied name. This is cosmetic, and has no effect on the structure of the table.

-h

Help gives a brief description of the command syntax [24, 25].

2.4.6.2 PARAMETERS

The following parameters make up a rule specification (as used in add, delete, insert, replace and append commands).

-p --protocol [!] *protocol*

The specified protocol can be one of *tcp*, *udp*, *icmp*, or *all*, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The

number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted.

-s --source [!] *address*[/*mask*]

Source specification. *Address* can be a network name, a hostname, a network IP address (with /mask), or a plain IP address. The *mask* can be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask. A "!" argument before the address specification inverts the sense of the address. The flag --src is an alias for this option.

-d --destination [!] *address*[/*mask*]

Destination specification. *Address* can be either a network name, a hostname etc. The flag --dst is an alias for this option.

-j --jump *target*

This specifies the target of the rule; i.e., what to do if the packet matches it.

-g --goto *chain*

This specifies that the processing should continue in a user specified chain. Unlike the --jump option return will not continue processing in this chain but instead in the chain that called us via –jump.

-i --in-interface [!] *name*

Name of an interface via which a packet was received (only for packets entering the INPUT, FORWARD and PREROUTING chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match. If this option is omitted, any interface name will match.

-o, --out-interface [!] *name*

Name of an interface via which a packet is going to be sent (for packets entering the FORWARD, OUTPUT and POSTROUTING chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match. If this option is omitted, any interface name will match.

[!] –f --fragment

This means that the rule only refers to second and further fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP type), such a packet will not match any rules which specify them. When the

"!" argument precedes the "-f" flag, the rule will only match head fragments, or unfragmented packets.

-c --set-counters *PKTS BYTES*

This enables the administrator to initialize the packet and byte counters of a rule (during INSERT, APPEND, REPLACE operations) [26].

2.4.7 How does a Packet Flow?

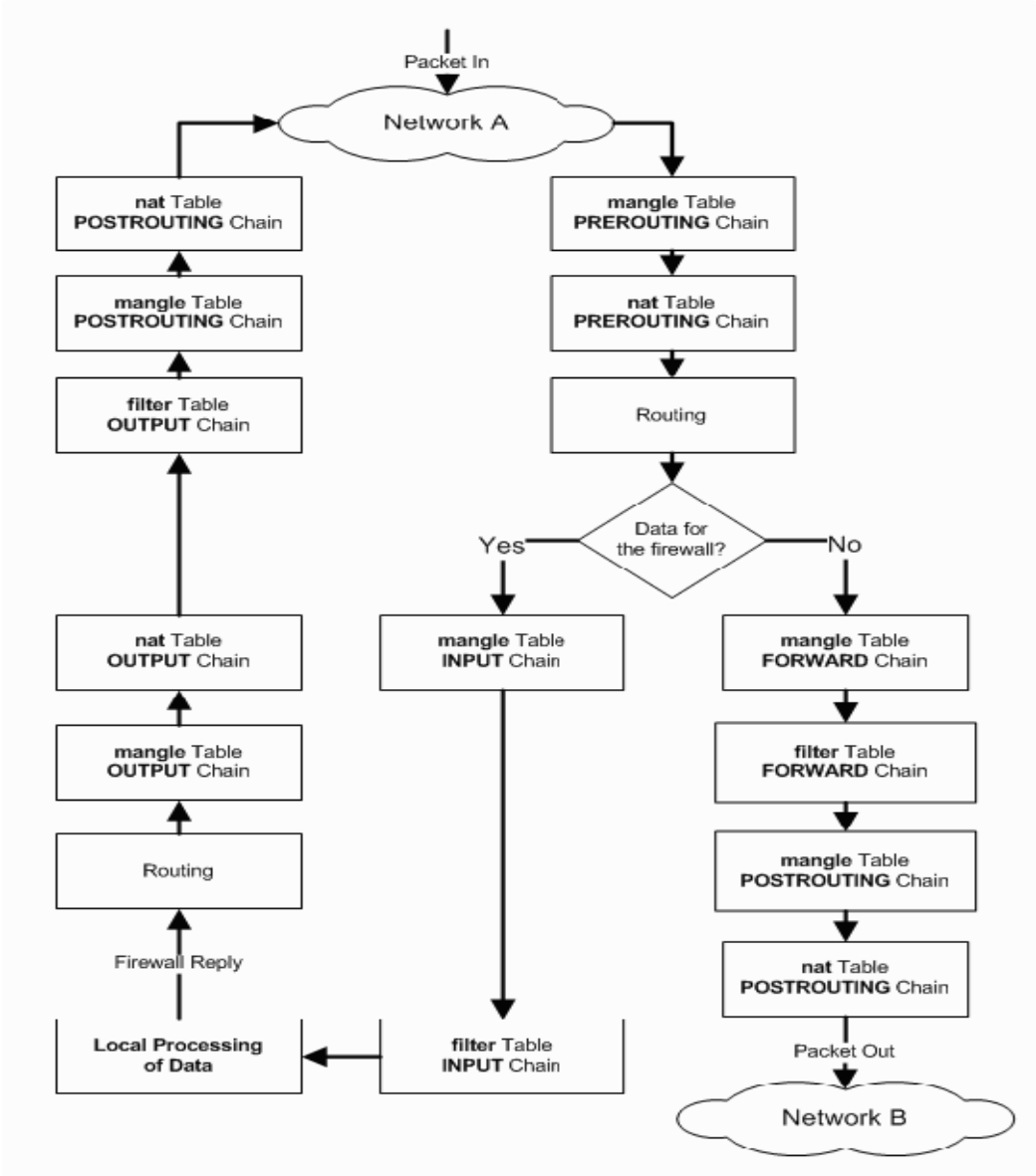


Figure 2.5: Packet Flow Diagram [20].

In Figure 2.5 a TCP packet from the Internet arrives at the firewall's interface on Network A to create a data connection. The packet is first examined by rules in the mangle table's PREROUTING chain, if any. It is then inspected by the rules in the

NAT table's PREROUTING chain to see whether the packet requires DNAT. It is then routed.

If the packet is destined for a protected network, then it is filtered by the rules in the FORWARD chain of the filter table and, if necessary, the packet undergoes SNAT in the POSTROUTING chain before arriving at Network B. When the destination server decides to reply, the packet undergoes the same sequence of steps. Both the FORWARD and POSTROUTING chains may be configured to implement quality of service (QoS) features in their mangle tables, but this is not usually done in SOHO environments.

If the packet is destined for the firewall itself, then it passes through the mangle table of the INPUT chain, if configured, before being filtered by the rules in the INPUT chain of the filter table before. If it successfully passes these tests then it is processed by the intended application on the firewall.

At some point, the firewall needs to reply. This reply is routed and inspected by the rules in the OUTPUT chain of the mangle table, if any. Next, the rules in the OUTPUT chain of the NAT table determine whether DNAT is required and the rules in the OUTPUT chain of the filter table are then inspected to help restrict unauthorized packets. Finally, before the packet is sent back to the Internet, SNAT and QoS mangling is done by the POSTROUTING chain [20].

Packet sniffing tools have been available from the early days of networked computing environments. The tools are powerful software, which facilitate troubleshooting for network administrators. However, in the hands of a malicious third party, they are a devastating hacking tool, which can be used to glean passwords and other sensitive information from a LAN.

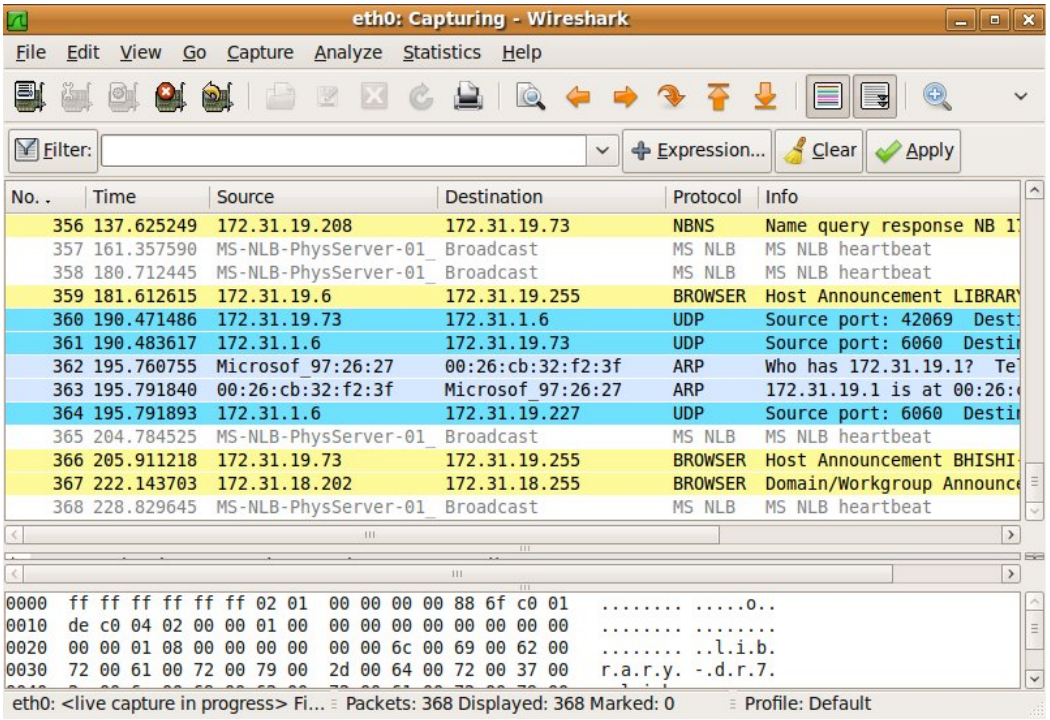
2.5 List of Packet Sniffing Tools

The following are some popular tools for administrating the network. These include some packet sniffing tools and display them in an organized manner.

2.5.1 Wireshark

The Wireshark is an open source network protocol analyzer, otherwise known as packet sniffer. It can capture live network traffic or read from a file and translate the data to be presented in a format the user can understand. Network Analyzers such as Wireshark are invaluable tools for administrators to diagnose and troubleshoot

problems with, but are also used by intruders to obtain unauthorized information. Wireshark isn't an intrusion detection system. It will not warn when someone does strange things on your network that he/she isn't allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on. Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn't send packets on the network or do other active things (except for name resolutions, but even that can be disabled). Distributions of Wireshark are available for a wide range of UNIX and Linux platforms as well as Windows. To actually capture the packets from the network requires a packet capture library like WinPcap (on Windows) or Libpcap (on Linux). Wireshark is sponsored by CACE, developers of the WinPcap library. The packet driver used will vary depending on the exact UNIX, Linux or Windows platform running Wireshark on. It has a GUI front-end, and many more information sorting and filtering options. It allows the user to see all traffic being passed over the network by putting the network interface into promiscuous mode. Given below in snapshot 2.1 the Wireshark in action. Wireshark is a development fork of the popular Ethereal Protocol Analyzer. Wireshark, like Ethereal, supports almost 700 protocols, more than most even know exist [27].



Snapshot 2.1: Wireshark - Network Protocol Analyzer

2.5.2 TCPdump

TCPdump is another powerful tool that allows us to sniff network packets and makes some statistical analysis out of those dumps. One major drawback to TCPdump is the size of the flat file containing the text output. But TCPdump allows us to precisely see all the traffic and enables us to create statistical monitoring scripts. Unlike Wireshark, it does not have GUI and is used on the command prompt. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached [28].

Chapter 3
Problem Formulation

3.1 Problem Statement

Information systems security is complicated not only by rapid growth in computer use and computer crime, but also by the complexity of computer networks. In addition, since absolute protection is not feasible, developing effective information systems security involves an often complicated set of trade-offs. The following consideration are (1) type and sensitivity of the information to be protected, (2) vulnerabilities of the computers and networks, (3) various threats, including hackers, thieves, disgruntled employees, competitors etc. Network security has become interwoven with standard network and system administration. Threats in the form of malicious hackers, self-propagating worms, denial of service attacks, and other nefarious security problems loom large for administrators. Of course, one of the building blocks of network security is a good firewall.

Administrator of an organization see some sites as offensive or bad, so the administrator wants that no one can access these sites. The easiest way that can be adopted is to use a dedicated firewall and block all those IP addresses that are frequent users of those sites or blocking of complete sites so that the user is unable to access the sites. Alone IDS and Antivirus is not sufficient for securing of network from the malicious attack. An IDS is used to detect many types of malicious network traffic and computer usage. This includes network attacks against vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware (viruses, trojan horses, and worms).An intrusion detection system (IDS) generally detects unwanted manipulations of computer systems, while a firewall is a dedicated appliance, or software running on another computer, which inspects network traffic passing through it, and denies or permits passage based on a set of rules. IDS is not a substitute for a Firewall instead an IDS works along with the firewall. Firewall can stop incoming requests to inherently insecure services, e.g. you can disallow rlogin, or RPC services such as NFS.They can control access to other services e.g. bar callers from certain IP

addresses, filter the service operations (both incoming and outgoing), e.g. stop FTP writes, hide information e.g. by only allowing access to certain directories or systems.

IPtables is a Linux based packet filtering firewall that has been used to allow/deny the network traffic on the basics of source IP, destination IP, port number, and protocol type. Work has been done on creating the virtual network environment using Microsoft virtual PC and capturing and analyzing of network packets using the most popular open source network protocol analyzer Wireshark and on the basis of analyzing the packets writing of script to block/allow the network traffic using IPtables and after blocking traffic further capturing and analyzing of live traffic using Wireshark.

Chapter 4

Implementation Details and Experimental Results

In this chapter work has been done on creating the virtual network environment using Microsoft virtual PC and then, capturing of Live traffic using the network protocol analyzer Wireshark and offline analyzing of these network data packets and after that on the basis of analyzing the packets writing of script to block/allow the network traffic using IPtables and after blocking traffic further capturing and analyzing of live traffic using Wireshark.

4.1 Implementation Setup

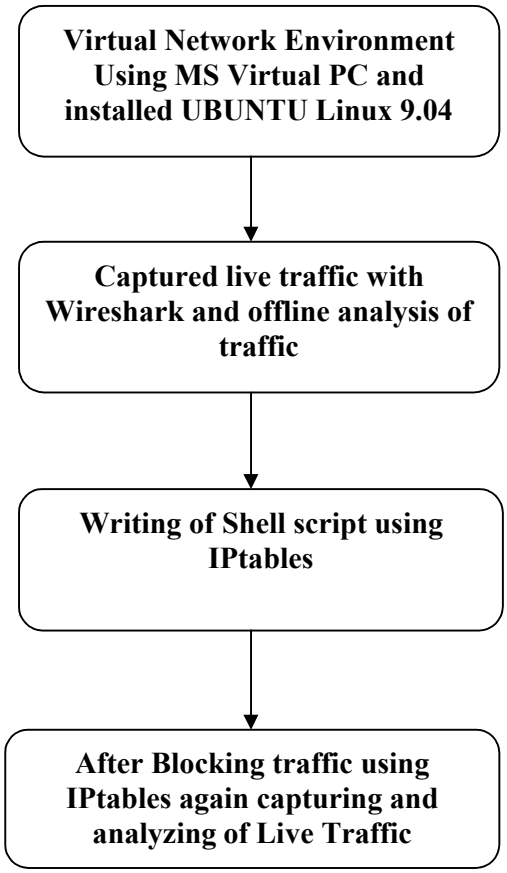


Figure 4.1: Implementation setup Diagram

4.2 Steps performed during Configuration and Implementation:

Step1: To establish a segregate network using virtualization. Microsoft Virtual PC SP1 is used to establish a segregate network and UBUNTU 9.04 operating system is installed on it.

Step2: Configuring the Wireshark under root privileges Applications > Add/Remove applications > All open source application > Wireshark.

Step3: Operating Wireshark (as root) and capturing network packets on eth0 interface.

Step4: IPtables was downloaded under the root privileges from www.netfilter.org and installed by implementing: ./configure, make, and make install command.

1) ./configure --prefix=/usr \
   --bindir=/sbin \
   --sbindir=/sbin \
   --libdir=/lib \
   --libexecdir=/lib \
   --with-pkgconfigdir=/usr/lib/pkgconfig &&
2) Make
3) Make install

--bindir=/sbin, --sbindir=/sbin Ensures all the executables goes in /sbin.

--libdir=/lib, --libexecdir=/lib Ensures all the libraries are in the /lib directory tree.

--with-pkgconfigdir=/usr/lib/pkgconfig Ensures all the pkgconfig files are in the standard location.

The list of Programs, Libraries, and Directories of IPtables

- **Installed Programs:**
  iptables,iptables-restore,iptables-save,iptables-xml,iptables-multi,ip6tables,ip6tables-restore, ip6tables-save, and ip6tables-multii
- **Installed Libraries:**
  libip4tc.so, libip6tc.so, libiptc.so, libxtables.so, and numerous modules in /lib/xtables/
- **Installed Directories:**
  /lib/xtables/xtables and /usr/include/libiptc

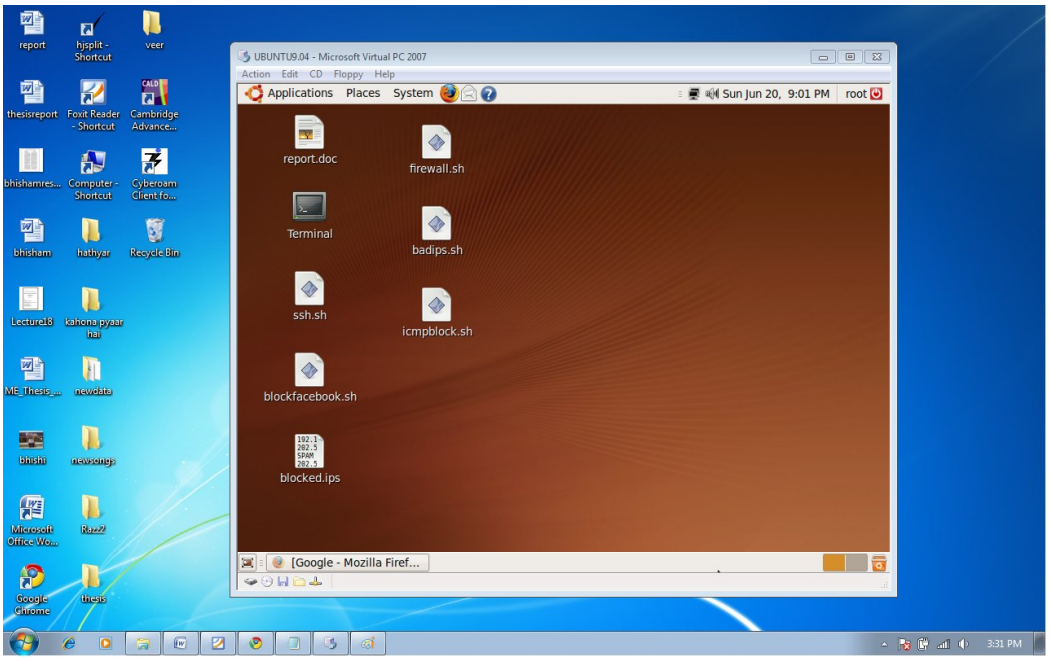
Step5: Writing of Shell script using IPtables.

Step6: After blocking traffic using IPtables again capturing and analyzing of Live Traffic using Wireshark.

4.3 Milestones covered and Experimental Results

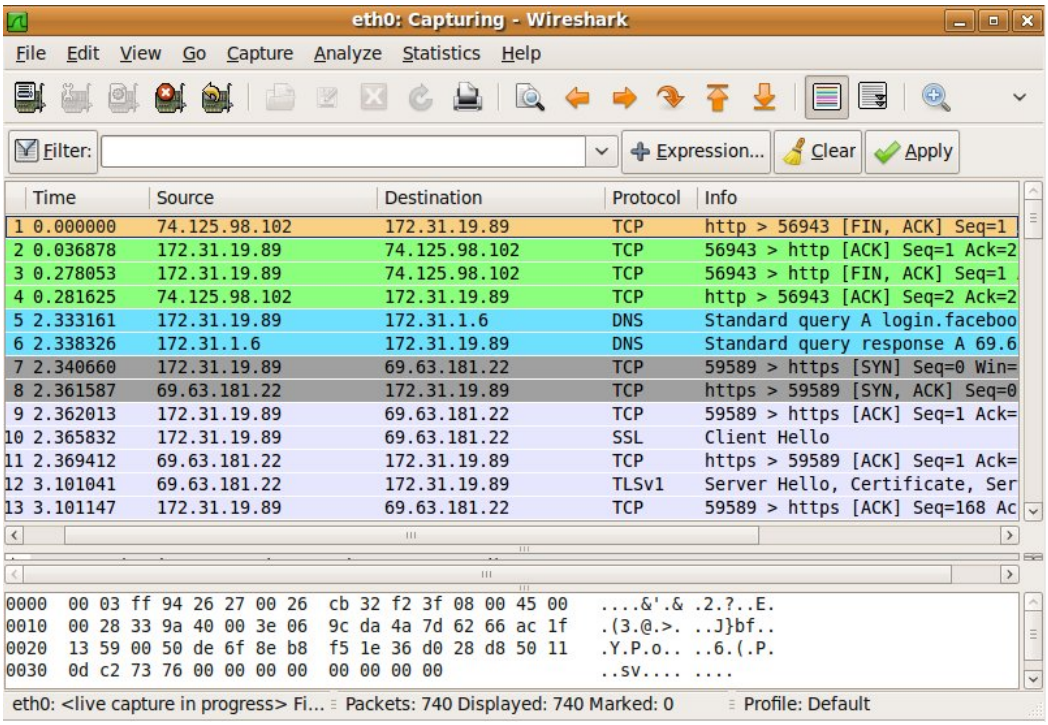
The following are the various tasks and milestones completed along with the results. All the results are checked on machine Intel® Core™ 2 Duo CPU T8100 @ 2.10 GHz with frequency 778 MHz along with 2 GB RAM space. The Operating System used was Ubuntu-9.04 Linux operating system which was installed using virtualization software.

4.3.1 Microsoft Virtual PC SP1 is used to establish a segregate network and UBUNTU 9.04 operating system is installed on it.

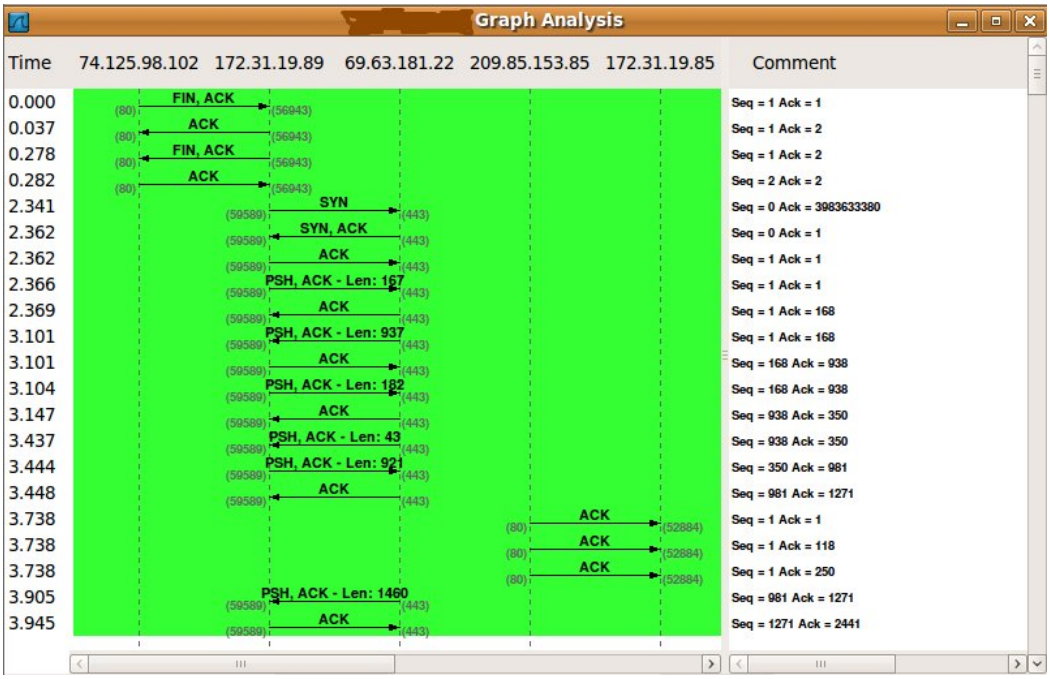


Snapshot 4.1: Virtual Network Environment

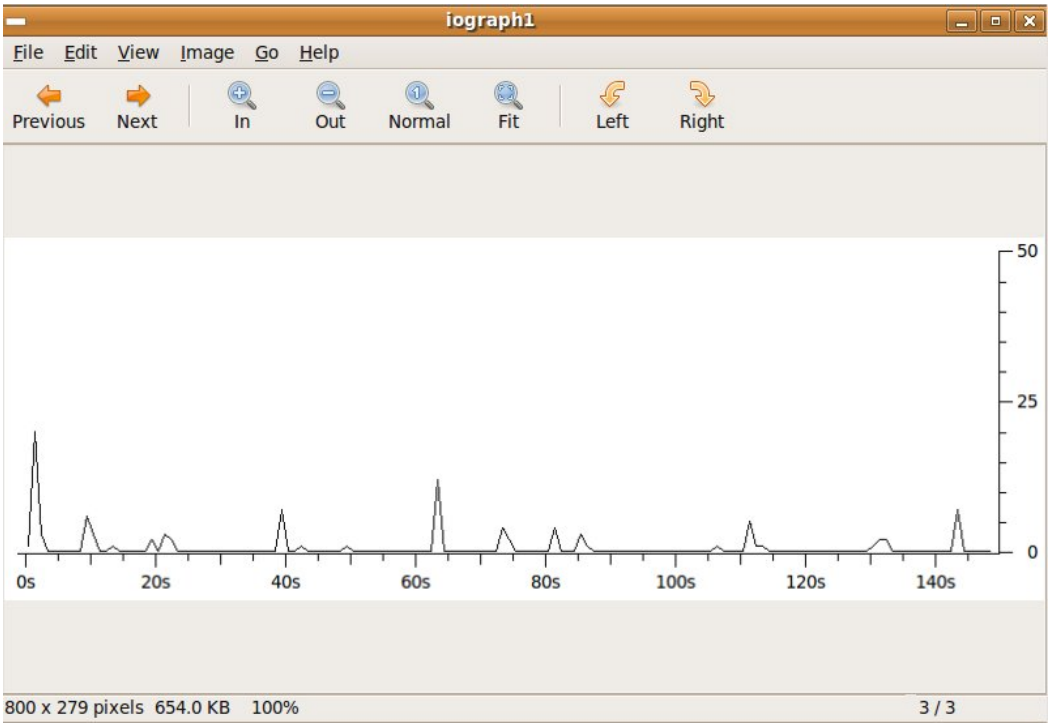
4.3.2 Capturing and analyzing of network packets using the most popular open source network protocol analyzer Wireshark.



Snapshot 4.2: Capture Packet using Wireshark



Snapshot 4.3: Wireshark TCP flow graph



Snapshot 4.4: Wireshark I/O graph

4.3.3 Module1. In this script work has been done to allow all the HTTP traffic on port number 80 without blocking any IP address. Firstly flushes all existing rules so to start with a clean state from which to add new rules. The -P switch sets the default policy on the specified chain. Setting of default policy on the INPUT chain to DROP. This means that if an incoming packet does not match one of the following rules it will be dropped. Similarly, here setting of the default policy on the FORWARD chain to DROP as not using the computer as a router so there should not be any packets passing through our computer and finally, setting of the default policy on the OUTPUT chain to ACCEPT as to allow all outgoing traffic. Here using of the -m switch to load a module (state). The state module is able to examine the state of a packet and determine if it is NEW, ESTABLISHED or RELATED. NEW refers to incoming packets that are new incoming connections that weren't initiated by the host system. ESTABLISHED and RELATED refers to incoming packets that are part of an already established connection or related to already established connection.

Script: httpallow.sh

```
#!/bin/bash
# flushes all rules for all chains
```

iptables -F

 # Set the default policies for the INPUT, OUTPUT and FORWARD chain.

iptables -P FORWARD DROP

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

 # Allow all loopback traffic

iptables -A INPUT -i lo -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT

 # Allow all incoming/outgoing HTTP traffic on port no. 80

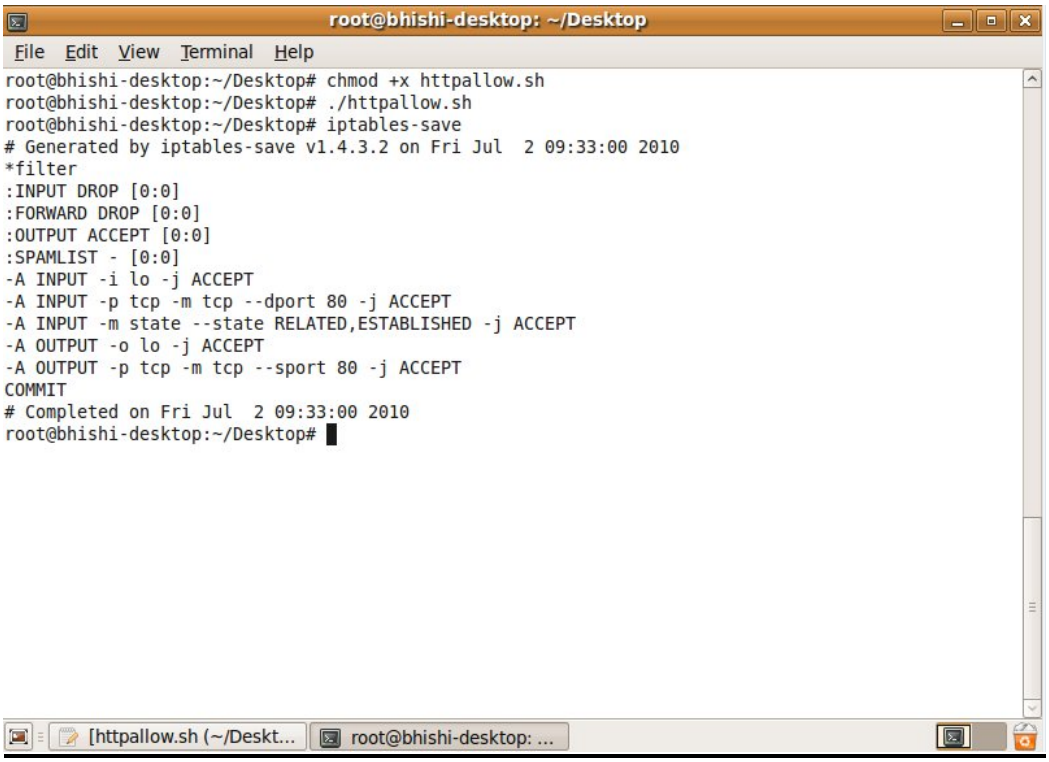
iptables -A INPUT -p tcp - -dport 8o -j ACCEPT

iptables -A OUTPUT -p tcp - -sport 8o -j ACCEPT

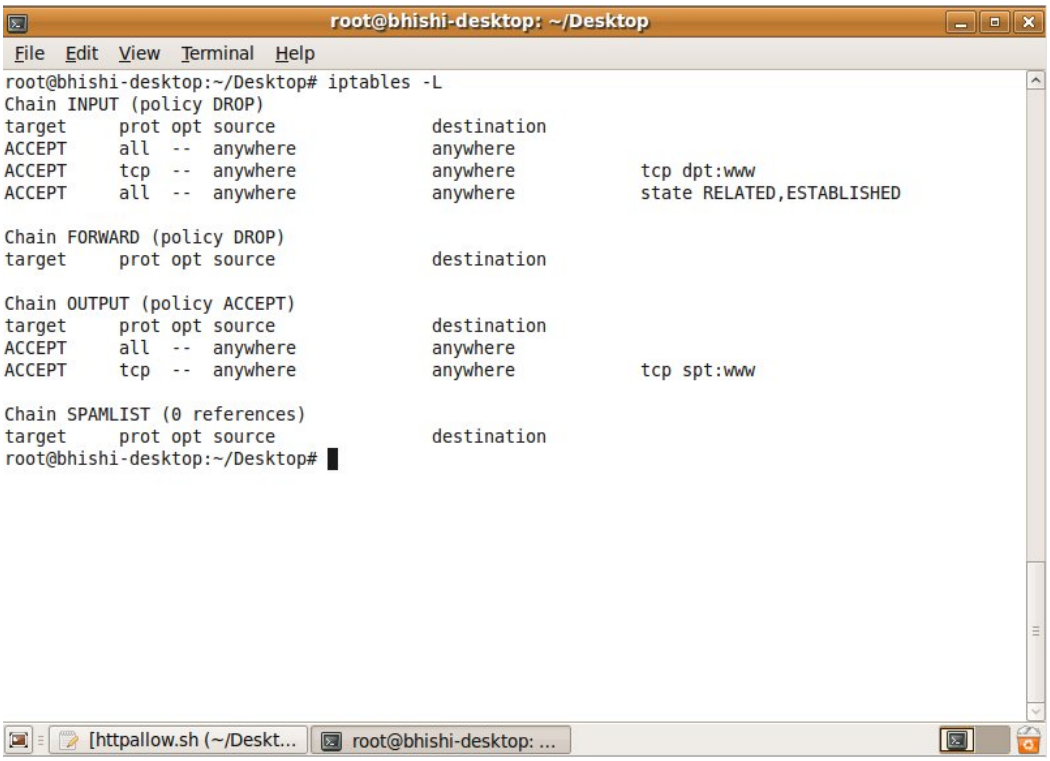
 # Allow returning traffic from locally initiated requests

iptables -A INPUT -m state - -state ESTABLISHED,RELATED -j ACCEPT

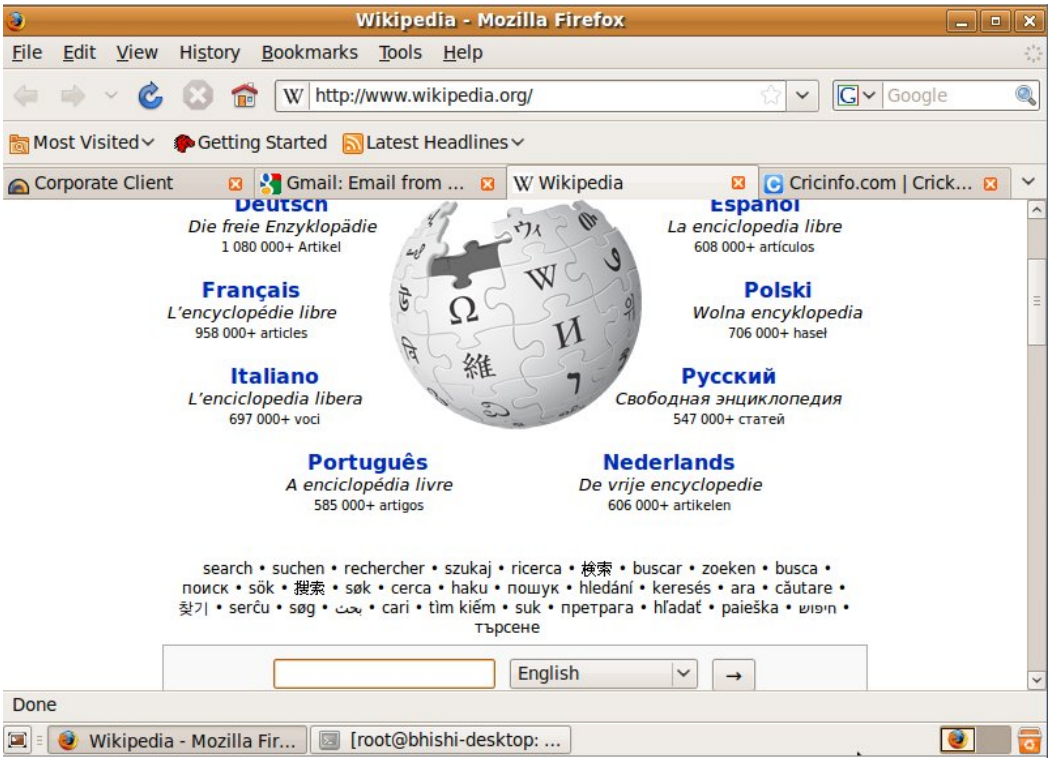
Output:



Snapshot 4.5: Running and Saving of IP Tables Script



Snapshot 4.6: Listing of IP Tables Rules



Snapshot 4.7: Allow HTTP Traffic

4.3.4 Module2. This module simply blocked HTTP traffic on TCP port no.80 on the Basis of specific URL's, Here blocking of social networking website that need to be banned in educational institution like www.facebook.com, www.myspace.com, and www.Bebo.com etc.

Facebook, Myspace, Bebo and Buzz allow students to contact people over the internet. They make 'friends' with people that they can't be sure are genuine and that can lead to problems. MySpace, Facebook and Bebo also encourage people (including children) to put photographs and other details about themselves into the system. Facebook is now the most popular social networking site worldwide, with Myspace second.

Many schools and public libraries in the US and the UK have begun to restrict access to Facebook and Myspace because it has become "such a haven for student gossip and malicious comments".

Script: httpblock.sh

```
#!/bin/bash
#
# Flush all current rules from iptables
iptables –F
#
# Set the default policies for the INPUT, OUTPUT and FORWARD chain.
#
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
# Allow all loopback traffic
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Blocking of All incoming/outgoing HTTP traffic on port no 80
iptables -A OUTPUT -p tcp -j DROP
# Blocking of Specific URL on the basis of their port no. and Domain name
#Blocking of www.Facebook.com accessed through port no 80
iptables -A OUTPUT -p tcp --dport 8o -d www.facebook.com -j DROP
#Blocking of www.Myspace.com accessed through port no 80
```

iptables -A OUTPUT -p tcp --dport 8o -d www.myspace.com -j DROP

#Blocking of www.bebo.com accessed through port no 80

iptables -A OUTPUT -p tcp --dport 8o -d www.bebo.com -j DROP

#Blocking of www.orkut.com accessed through port no 80

iptables -A OUTPUT -p tcp --dport 8o -d www.orkut.com -j DROP

Allow returning traffic from locally initiated requests

iptables -A INPUT -m state - -state ESTABLISHED,RELATED -j ACCEPT

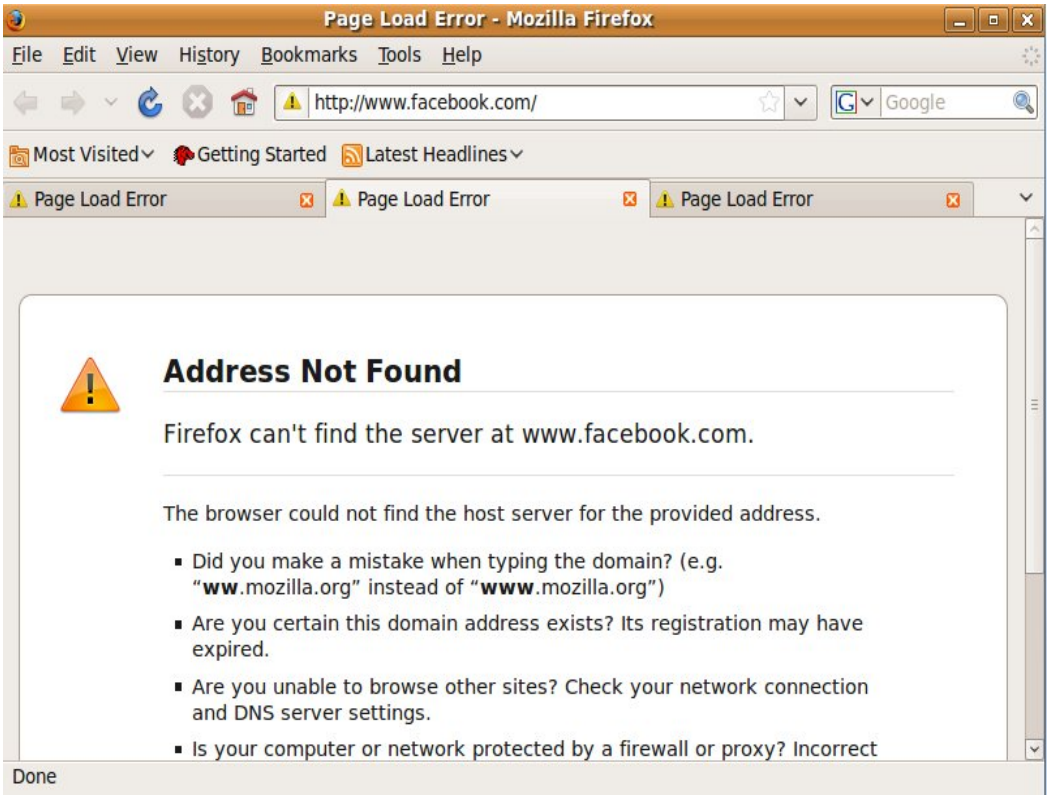
#

 # List rules

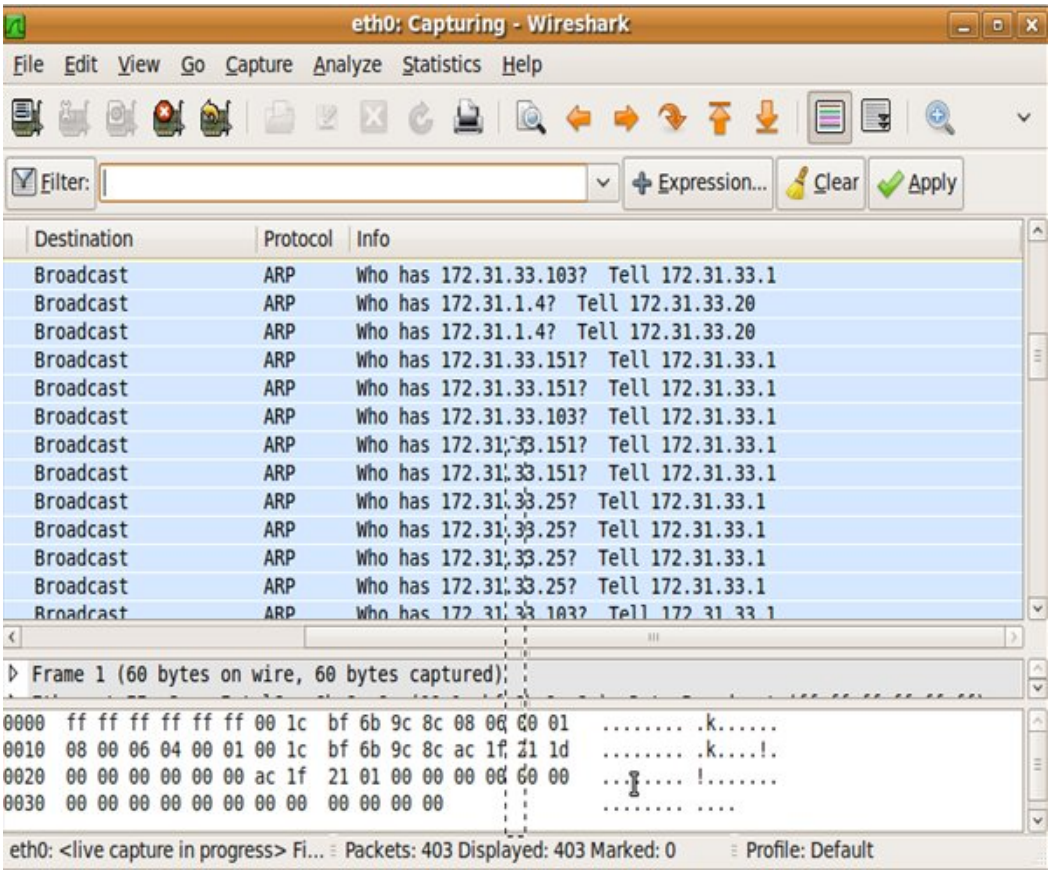
#

iptables -L –v

Output



Snapshot 4.8: Blocking of HTTP Traffic



Snapshot 4.9: Captured Packets after blocking HTTP traffic

4.3.5 Module3. Blocking of Spam mails coming from specific IP address to secure the network so that unauthorized user was unable to access the resources of the system. SPAM is one giant pain in the neck. There are usually fifty other things you would prefer to be doing than finding new ways to defeat spam. Here, creation of new chain SPAMLIST in which all the rules are appended through which blocking of spam mails coming from bad IP's.

Blocked.ips is a file in which list of bad IP's is mentioned.

Script: blockspamip.sh

```
#!/bin/bash
# IPtables IP/subnet block script
IPT=/sbin/iptables
SPAMLIST="spamlist"
SPAMDROPMSG="SPAM LIST DROP"
BADIPS=$(grep  -Ev "^#|^$" "/root/blocked.ips")
```

 # create a new IPtables list

$IPT -N $SPAMLIST

for ipblock in $BADIPS

do

  $IPT -A $SPAMLIST -s $ipblock -j LOG --log-prefix "$SPAMDROPMSG"

  $IPT -A $SPAMLIST -s $ipblock -j DROP

done

$IPT -I INPUT -j $SPAMLIST

$IPT -I OUTPUT -j $SPAMLIST

$IPT -I FORWARD -j $SPAMLIST

 # List rules

#

iptables -L –v

Blocked.ips

192.168.1.0/24

Vsnloard.vsnl.net.in

202.54.1.2

SPAM

202.5.1.2

Personal information stealers (Big Brother):

doubleclick.net

216.73.93.8

The SPAMMERs never stop to find new ways to SPAM

Messageaway.com

74.86.203.162

sofcom.com

98.124.199.1

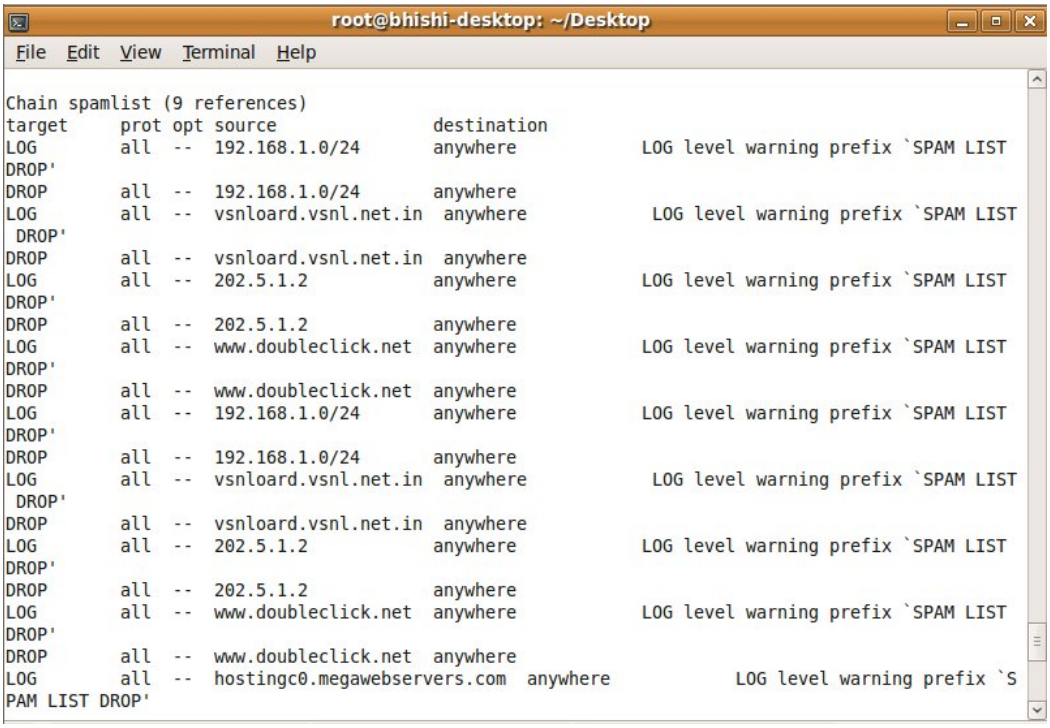
mp3za.ru

203.110.240.22

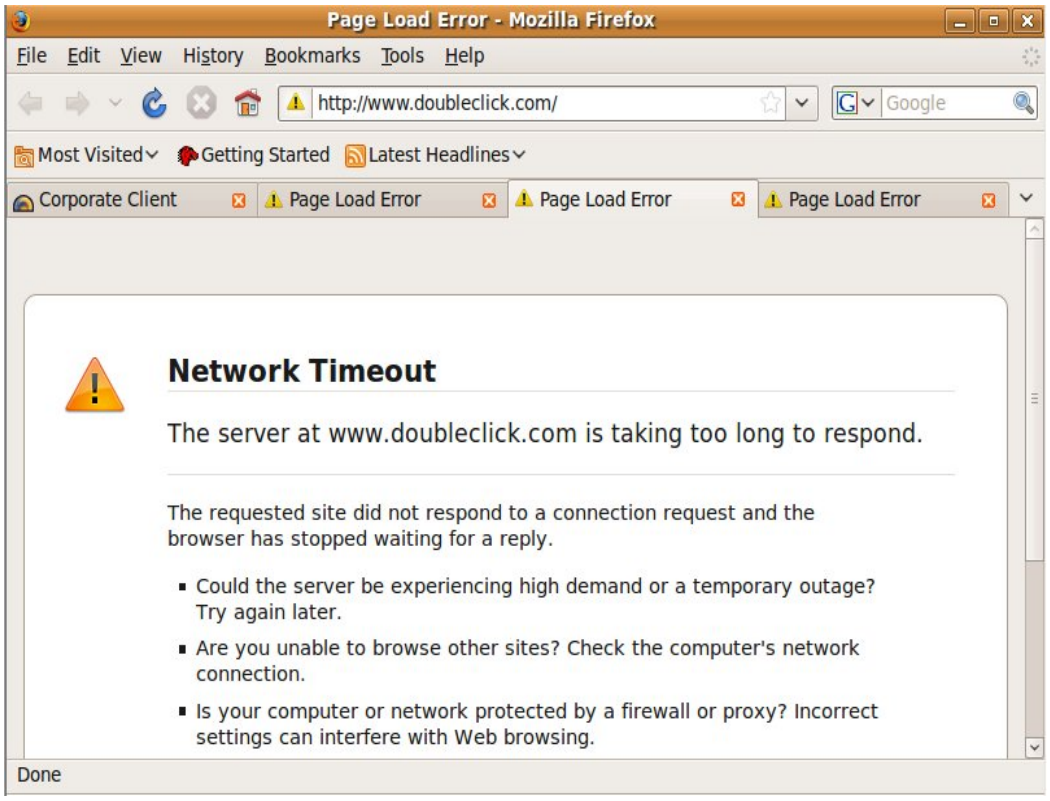
free4all.com

64.95.64.198

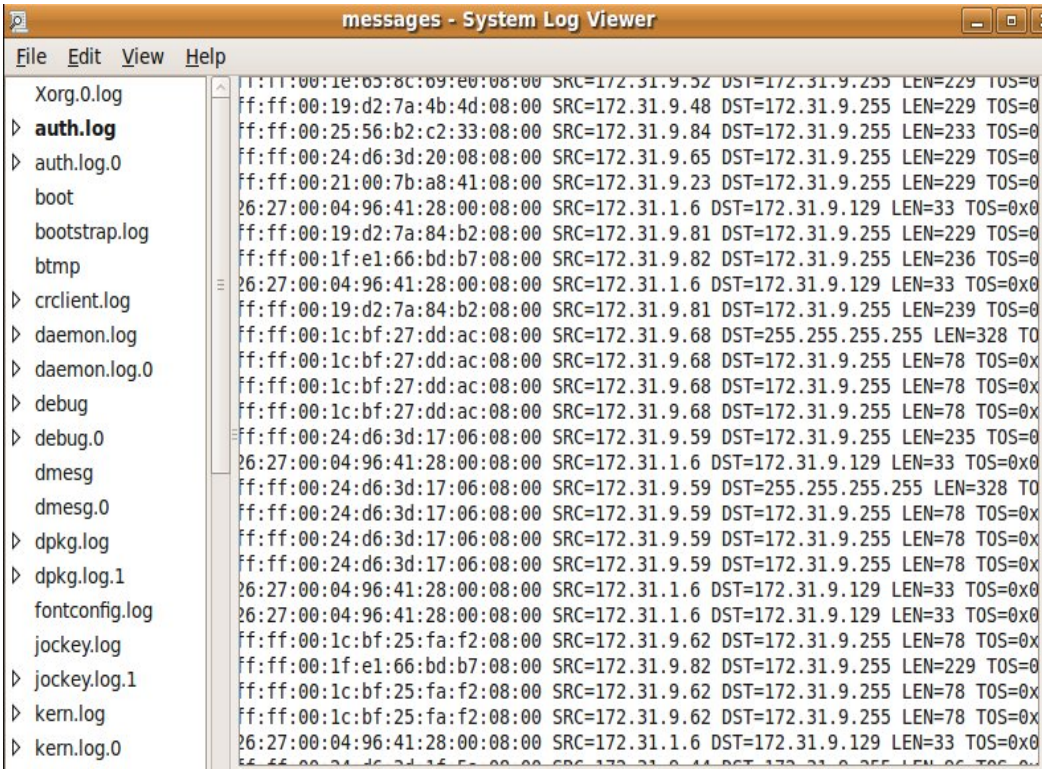
Output



Snapshot 4.10: Listing of Spam list chain



Snapshot 4.11: Denying of spam sites



Snapshot 4.12: System Log

4.3.6 Module4. Blocking of ICMP packet so that unauthorized user is unable to ping the system. Several Web sites block ICMP traffic due to DDoS attacks. A common example of "bad" ICMP is to allow any ICMP traffic from unknown sources onto your trusted networks. For example, if you allow ICMP redirects, you leave your Internet hosts susceptible to having their traffic inadvertently routed to the wrong location. This could result in a DoS is the best case (because the traffic never makes it to the hosts that are requesting data). To address this, it is generally a good idea to block ICMP traffic, in particular between authorized and unauthorized networks. ICMP messages themselves are also susceptible to manipulation. Perhaps the most well known of this kind of manipulation is known as the "ping of death," which transmitted a message that exceeded the 65,535-byte limit of the IP protocol, which would cause many target hosts to crash, resulting in a DoS.

Script: icmpblock.sh

```
#!/bin/bash
# flushes all rules for all chains
iptables -F
# set the default policies
```

iptables -P FORWARD DROP

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

#

In the mentioned method best thing is to drop the ICMP packets, by doing this not giving any clue to hacker whether the system is alive or not. Where as if option is reject definitely hacker will come to know that ICMP packets are blocked and the system is live.

#

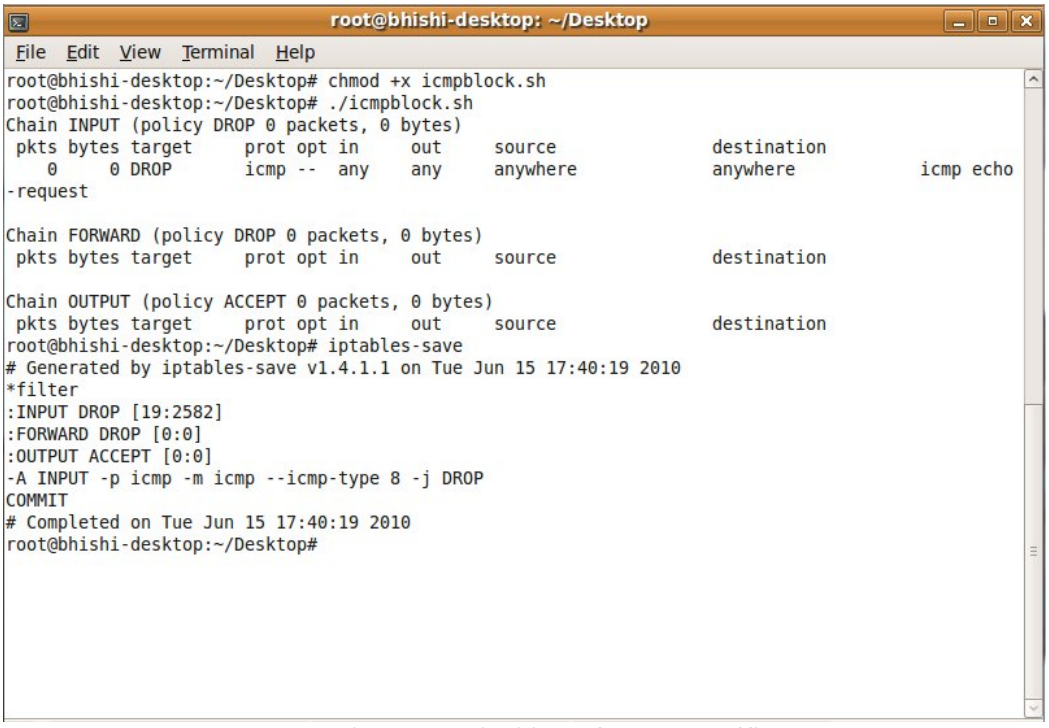
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

 # List rules

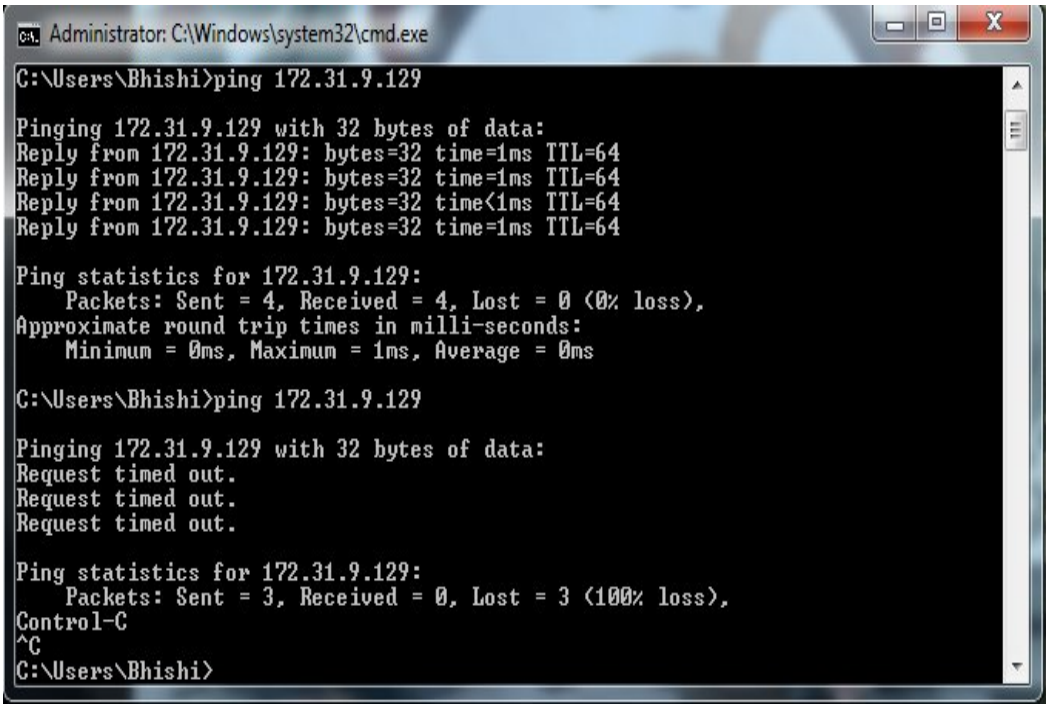
#

iptables -L -v

Output



Snapshot4.13: Blocking of ICMP Traffic



Snapshot4.14: Status of Ping Command

4.3.7 Module5. In this module simply stop incoming/outgoing SMTP traffic to protect the system from the various types of attacks like Phishing, Hoaxes, and Trojans. Phishing is a scam, where a stranger sends an email which appears as if it is from a trusted organization to a normal user to get his personal and financial information. For example, when you receive a mail from a bank to update your personal bank account information and when you click on the link to update the information a separate window opens which looks like an original bank site, where it asks for account information, password and other details. When you enter the information and press enter it will go to the hands of strangers and not to the bank site. Hoax is an attempt to make the person believe something which is false as true. It also defined as an attempt to deliberately spread fear, doubt among the users.

Script: smtpblock.sh

```
#!/bin/bash
#
# flushes all rules for all chains
#
 iptables -F
```

#

 # stopping of reading and writing of Emails

iptables -A INPUT -p tcp --dport 25 -j DROP

iptables -A INPUT -p udp --dport 25 -j DROP

iptables -A INPUT -p udp --dport 110 -j DROP

iptables -A OUTPUT -p tcp --sport 25 -j DROP

iptables -A OUTPUT -p udp --sport 25 -j DROP

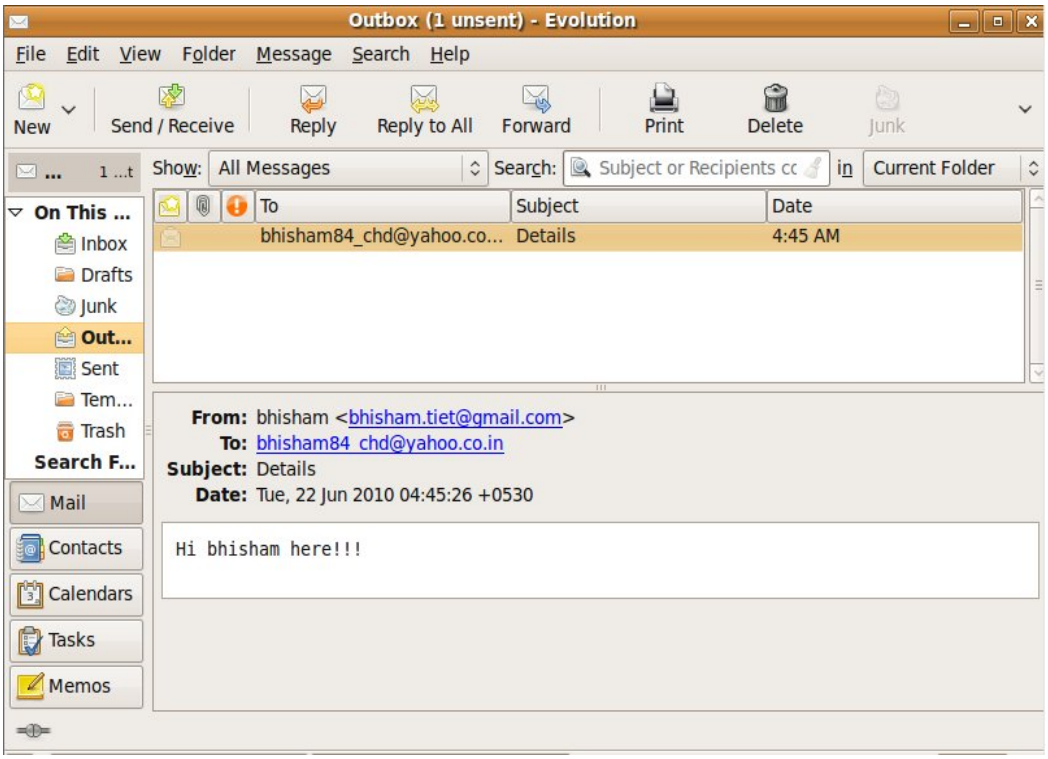
iptables -A OUTPUT -p udp --sport 110 -j DROP

 # List rules

#

iptables -L –v

Output



Snapshot 4.15: Deny SMTP Traffic

4.3.8 Module6. In this module work has been done for the design and development of Host based Firewall. Firstly flushes all the rules after that packets move freely if generated by the machine itself indicated by 'lo' or local host. Then allow packets to

leave the machine on our network card eth0. Next, start to prevent some of the most common network vulnerabilities. The first one is spoofing, or the ability to make packets appears as if they're coming from your machine or your network. As it is common practice to be a bit more tolerant toward those on your own network, that intent on breaking in will first try the spoofing routine. Next, block any bad packets - period. Another way of attacking a network is to overwhelm it with junk. Here control this. Next stop Xmas Tree type scanning, null scan, syn flood, ping flood attack etc.

The second part of the firewall deals with allowing and controlling access to certain services running on the machine. The comments below each rule show which services are dealt with each one. Providing of access control to both tcp and udp traffic. Some services are offered to the public and some are not.
Finally with the last rule drop all packets that aren't destined for the allowed ports.

Script: firewall.sh

```
#!/bin/bash
IPTABLES=/sbin/iptables
# start by flushing the rules
iptables  -F
# allow packets coming from the machine
iptables  -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# allow outgoing traffic
iptables -A OUTPUT -o etho -j ACCEPT
**********************************
 **********Block **********
**********************************
# block spoofing
iptables -A INPUT -s 127.0.0.0/8 -i ! lo -j DROP
iptables -A INPUT -s 192.168.0.3 -j DROP
# Stop bad packets
iptables -A INPUT -m state --state INVALID -j DROP
# NMAP FIN/URG/PSH
iptables -A INPUT -i etho -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
```

 # Stop Xmas Tree type scanning

iptables -A INPUT -i etho -p tcp --tcp-flags ALL ALL -j DROP

iptables -A INPUT -i etho -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

 # Stop null scanning

iptables -A INPUT -i etho -p tcp --tcp-flags ALL NONE -j DROP

 # SYN/RST

iptables -A INPUT -i etho -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

 # SYN/FIN

iptables -A INPUT -i etho -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

 # Stop sync flood

iptables -N SYNFLOOD

iptables -A SYNFLOOD -p tcp --syn -m limit --limit 1/s -j RETURN

iptables -A SYNFLOOD -p tcp -j REJECT --reject-with tcp-reset

iptables -A INPUT -p tcp -m state --state NEW -j SYNFLOOD

 # Stop ping flood attack

iptables -N PING

iptables -A PING -p icmp --icmp-type echo-request -m limit --limit 1/second -j RETURN

iptables -A PING -p icmp -j REJECT

iptables -I INPUT -p icmp --icmp-type echo-request -m state --state NEW –j PING

****

**********Allow **********

****

 # tcp ports

 # smtp

iptables -A INPUT -p tcp -m tcp --dport 25 -j ACCEPT

 # http

iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

 # pop3

iptables -A INPUT -p tcp -m tcp --dport 110 -j ACCEPT

 # imap

iptables -A INPUT -p tcp -m tcp --dport 143 -j ACCEPT

 # ldap

iptables -A INPUT -p tcp -m tcp --dport 389 -j ACCEPT

 # https

iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT

 # smtp over SSL

iptables -A INPUT -p tcp -m tcp --dport 465 -j ACCEPT

 # line printer spooler

iptables -A INPUT -p tcp -m tcp --dport 515 -j ACCEPT

 # cups

iptables -A INPUT -p tcp -m tcp --dport 631 -j ACCEPT

 # udp ports

 # DNS

iptables -A INPUT -p udp -m udp --dport 53 -j ACCEPT

 # DHCP

iptables -A INPUT -p udp -m udp --dport 67:68 -j ACCEPT

 # NTP

iptables -A INPUT -p udp -m udp --dport 123 -j ACCEPT

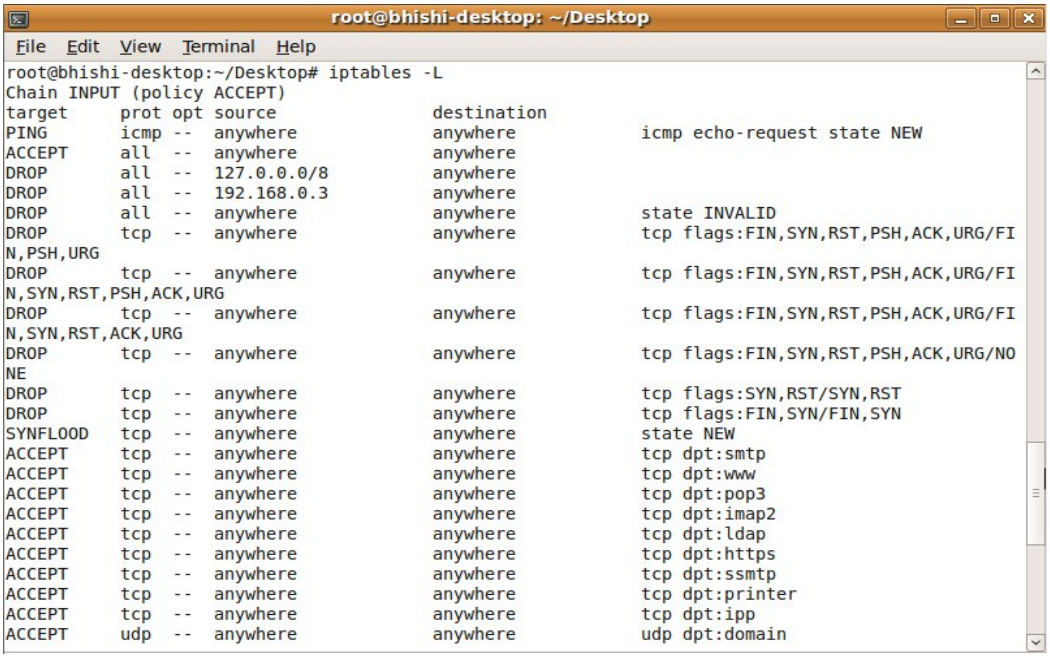
 # SNMP

iptables -A INPUT -p udp -m udp --dport 161:162 -j ACCEPT

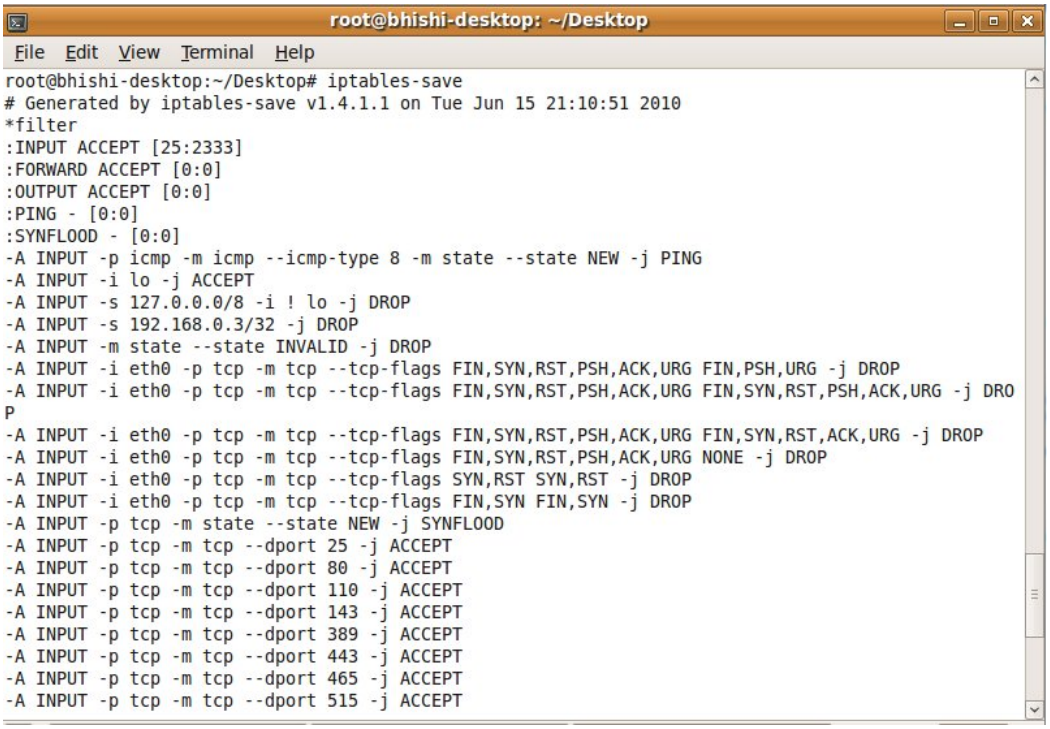
#Finally - drop the rest

iptables -A INPUT -p tcp --syn -j DROP

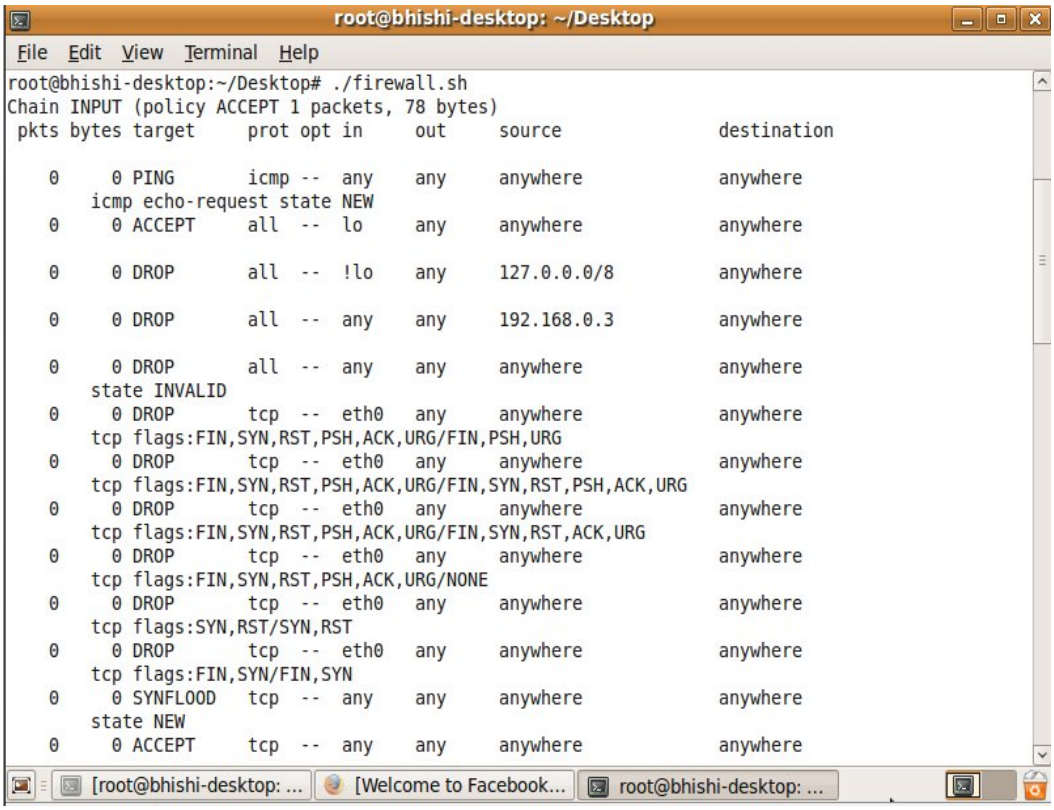
Output



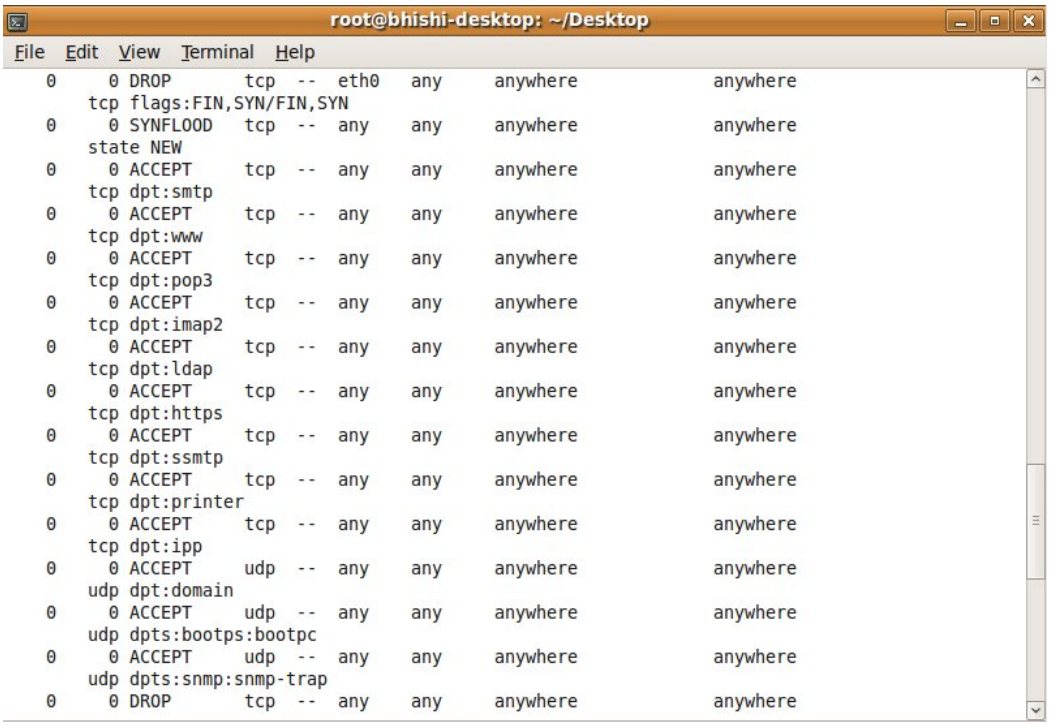
Snapshot 4.16: Listing of IP Tables rules



Snapshot4.17: Saving rules in IP Tables



Snapshot 4.18: Running of Firewall



Snapshot 4.19: Running of Firewall (Cont.)

4.3.9 Module7. In this module work has been done on blocking of P2P file sharing traffic. Peer-to-Peer (P2P) applications impede network traffic of businesses, governments, education, and the Internet infrastructure itself. These applications consume vast amounts of network resources, and prevent mission critical applications from accessing the network. In addition, hackers seek to exploit P2P applications to access and attack the large install base, presenting serious security vulnerabilities to systems and networks. These applications also pose a serious legal issue as users download copyrighted material, placing access providers in a difficult legal situation. As a result, these applications create a logistic, security, and legal nightmare for network administrators on high-speed networks. To protect networks from excessive bandwidth consumption and malicious attacks, work has been done on writing the IPtables script to control Peer-to-Peer applications and network traffic.

IPP2P is a module that is required for blocking the P2P traffic using IPtables:

- IPP2P

IPP2P is a netfilter extension to identify P2P file sharing traffic. The main goal of IPP2P was giving the administrator a dynamic tool to filter the traffic in an intelligent way. So it doesn't aim at prohibiting all P2P traffic but make it possible to shape this traffic to a given rate. For this purpose IPP2P searches the payload of TCP packets for signaling patterns of P2P networks. IPP2P works together with connection tracking and connection marking - in that way you can catch the bigger part of all P2P packets and limit the bandwidth rate.

The name IPP2P comes on the one side from IP (internet protocol) and on the other from P2P (short for peer-to-peer) [29].

The following table shows a lineup of all module options available for IPP2P.

Table 4.1: IPP2P Options [29].

option	P2P network	protocol
--edk	eDonkey, eMule, Kademlia	TCP and UDP
--kazaa	KaZaA, FastTrack	TCP and UDP
--gnu	Gnutella	TCP and UDP
--dc	Direct Connect	TCP only
--bit	BitTorrent, extended BT	TCP and UDP
--apple	AppleJuice	TCP only
--winmx	WinMX	TCP only
--soul	SoulSeek	TCP only
--ares	Ares, AresLite	TCP only

4.3.9.1 Various steps required for installation of IPP2P module.

Step1: Download ipp2p-0.8.2.tar.gz from ipp2p.org

Step2: Decompression tar –xzf ipp2p-0.8.2.tar.gz

Step3: Open Makefile in editor and setting of IPtables source.

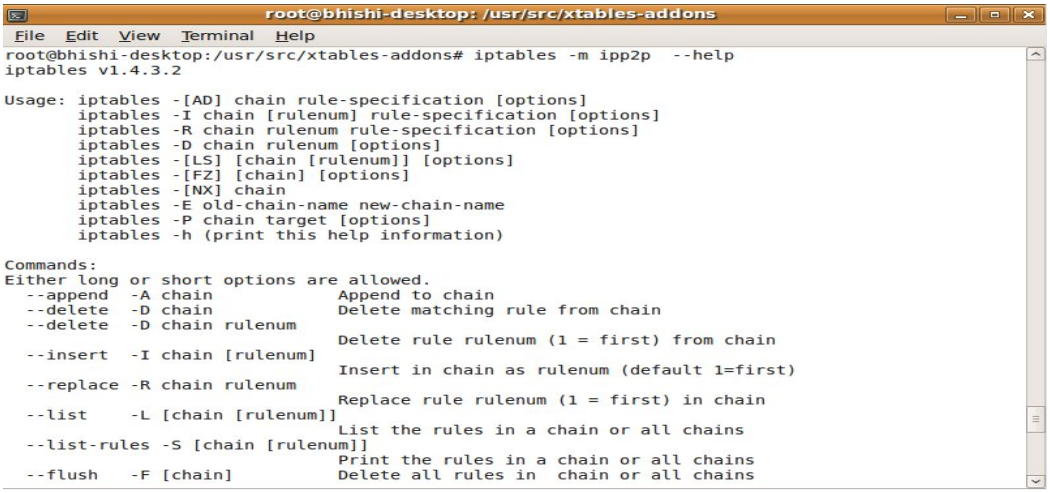
Step4: Compile IPP2P by typing "make" in console.

Step5: After compilation two files are created ipt_ipp2p.ko and libipt_ipp2p.so.

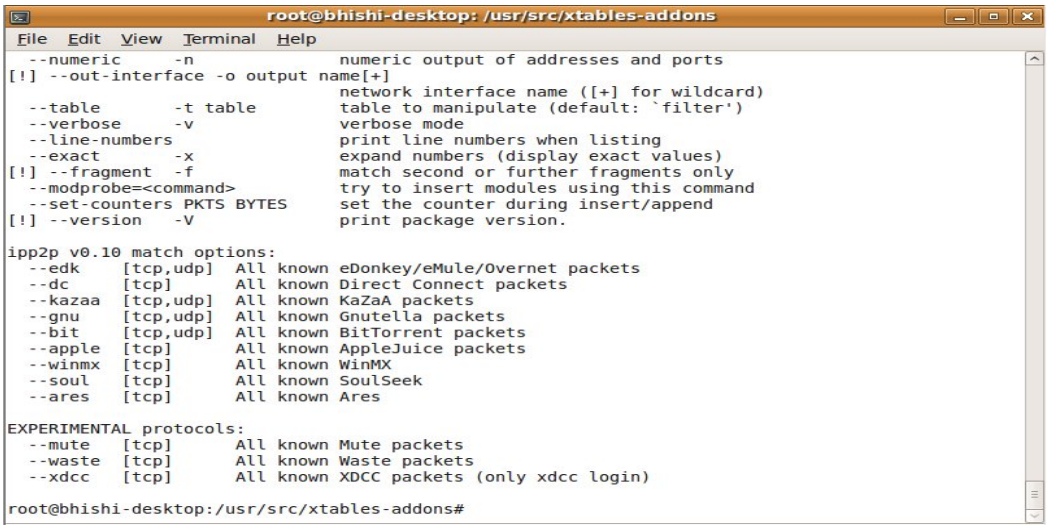
Step6: Copy of libipt_ipp2p.so to IPtables library directory /usr/lib/iptables/.

Step7: Testing of IPP2P exists by entering "iptables -m ipp2p --help"

Step8: Finally load the kernel module to be able to create a firewall rule using IPP2P "insmod ipt_ipp2p.ko" to load the module.



Snapshot 4.20: Testing of IPP2P module.

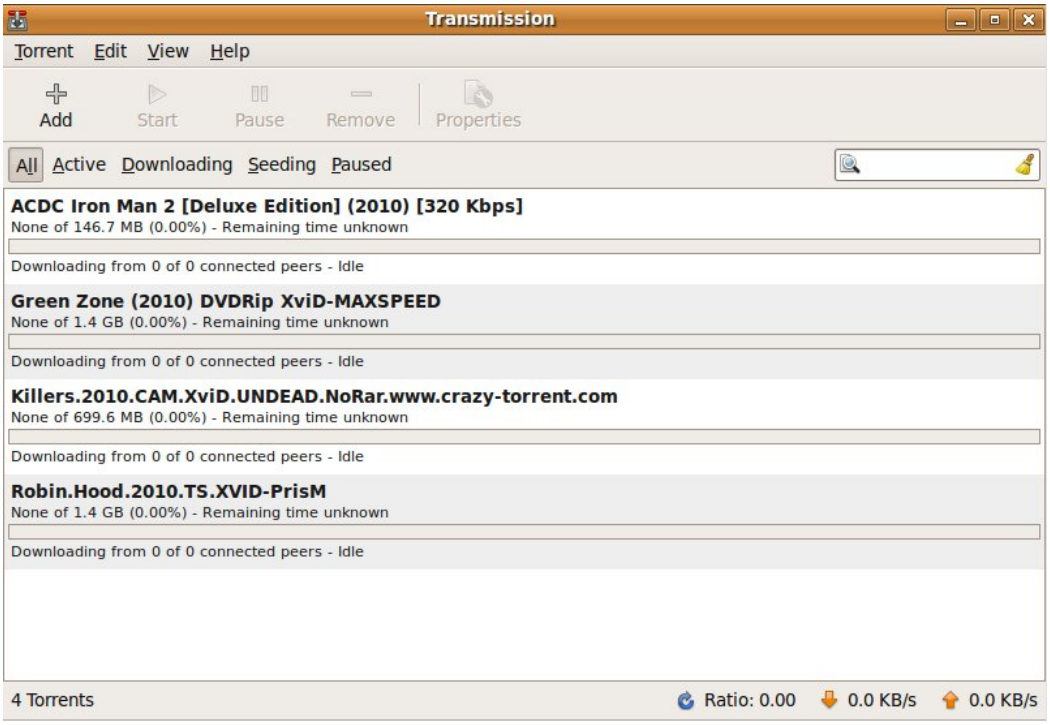


Snapshot 4.21: Testing of IPP2P module (Cont.).

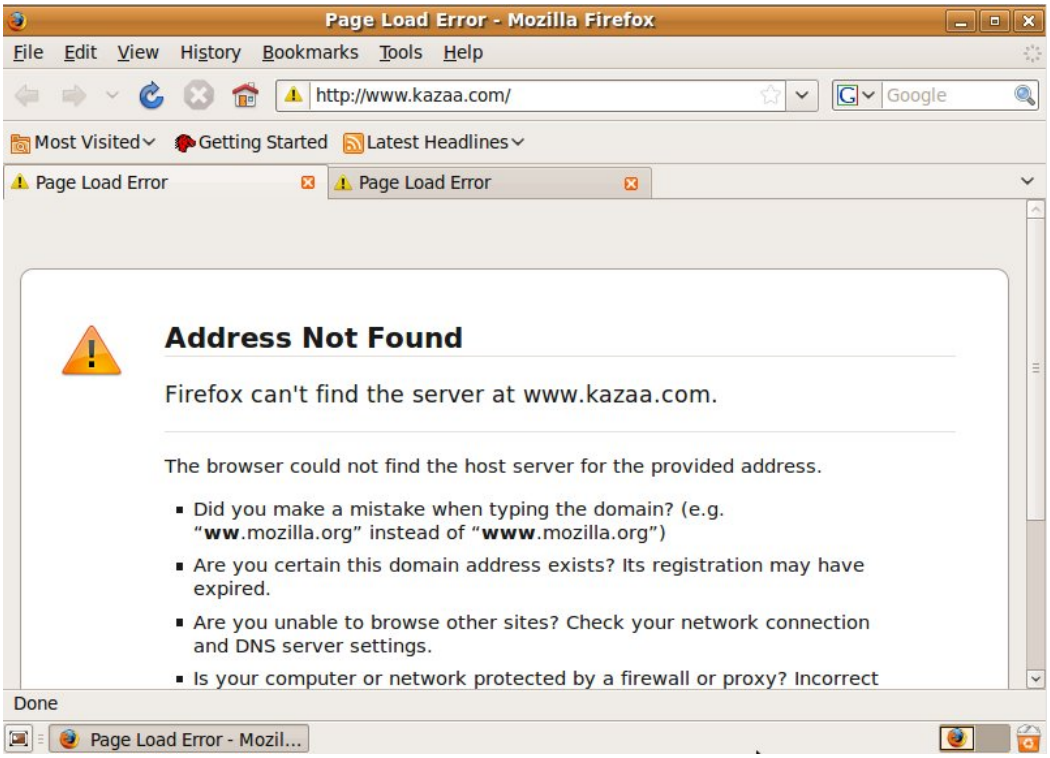
Script: p2pblock.sh

```
#!/bin/bash
IPTABLES=/sbin/iptables
# start by flushing the rules
iptables  -F
# allow packets coming from the machine
iptables  -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# allow outgoing traffic
iptables -A OUTPUT -o etho -j ACCEPT
# Block P2P Traffic
iptables -A FORWARD -p tcp -m ipp2p --edk -j DROP
iptables -A FORWARD -p udp -m ipp2p --edk -j DROP
iptables -A FORWARD -p tcp -m ipp2p --dc -j DROP
iptables -A FORWARD -p tcp -m ipp2p --kazaa -j DROP
iptables -A FORWARD -p udp -m ipp2p --kazaa -j DROP
iptables -A FORWARD -p tcp -m ipp2p --gnu -j DROP
iptables -A FORWARD -p udp -m ipp2p --gnu -j DROP
iptables -A FORWARD -p tcp -m ipp2p --bit -j DROP
iptables -A FORWARD -p udp -m ipp2p --bit -j DROP
iptables -A FORWARD -p tcp -m ipp2p --apple -j DROP
iptables -A FORWARD -p tcp -m ipp2p --winmx -j DROP
iptables -A FORWARD -p tcp -m ipp2p --soul -j DROP
iptables -A FORWARD -p tcp -m ipp2p --ares -j DROP
#Listing of Rules
iptables –L -v
```

Output



Snapshot 4.22: Blocking of Transmission Bit Torrent



Snapshot 4.23: Blocking of P2P sharing

Chapter 5
Conclusion and Future Scope

5.1 Conclusion

For a Linux system connected to a network, a firewall is the essential defense mechanism that allows only legitimate network traffic in and out of the system and disallows everything else. To determine whether the network traffic is legitimate or not, a firewall relies on a set of rules it contains that are predefined by a network or system administrator. These rules tell the firewall whether to consider as legitimate and what to do with the network traffic coming from a certain source, going to a certain destination, or having a certain protocol type.

Linux has become extremely popular in the IT industry because of its robustness, reliability, flexibility, and seemingly unlimited scope for customization. Linux has many inbuilt capabilities that let the developer customize its tools, behavior, and appearance according to his needs without requiring expensive third-party tools. One such inbuilt capability is firewall configuration for Linux systems on a network, be they systems connected to the Internet or a LAN, servers, or proxy servers interfacing between a LAN and the Internet. This capability can be put to use with the help of the Netfilter/IPtables IP packet filtering system that comes integrated with Linux kernels. The connection-tracking feature of IPtables is a very useful thing. It can be used to prevent most TCP hijackings for non-IP Masqueraded clients that suffer from poor TCP sequence number randomization. Similarly, it can be used to prevent UDP packet hijacking in the same way. This functionality can also prevent attackers from injecting spurious ICMP packets for cracking and probing. Packets can be matched based on MAC address, the local process's UID, Time to Live (TTL). These allow better detection and rejection of interlopers trying to inject packets or scan a system.

In this thesis, work has been done on capturing the live traffic using the network protocol analyzer Wireshark and on the basics of analyzed data packets further explored and designed the script using IPtables to allow/deny the network traffic on the basics of the IP address of the computer sending the packets, the IP address of the computer receiving the packets, the type of packet (TCP, UDP, etc.), The port

number, and URL's etc. This enables us to protect our system from a wide variety of hazards, including service attacks and hack attempts.

The script discussed here can be used for the purpose of network Security.

- Web traffic sent on HTTP can be analyzed.
- Denying of ICMP, SMTP data packets.
- Configuring of host based packet filtering firewall to deny various type of attacks like spoofing, Stop bad packets, Stop Xmas Tree type scanning, null scanning, syn flood and, ping flood attack etc.
- Deny P2P file sharing traffic.

5.2 Future Scope

The Netfilter/IPtables IP packet filtering system is the latest among Linux packet filtering solutions like ipfwadm and ipchains and is also the first one to be integrated into the Linux kernel. The Netfilter/IPtables system is ideal for Linux system administrators, network administrators, and home users who want to configure firewalls according to their specific needs, save money on firewall solutions, and have total control over IP packet filtering.

In this thesis work it could not experiment the advanced features of IPtables such as NAT, IP masquerading, packet redirect, IPtables has the ability to REDIRECT packets like IP Chains does, however it also has a generalized DNAT feature that allows arbitrary changing of the destination IP address and port number. Thus, it can actually disguise where packets of a given service go. This has uses everywhere from Honey Pots and Tarpits to enforcing the use of a given proxy server for web caching.

References

[1] Gunter Schafer, “Network Security Tutorial”, May 2003, Anchorage, Alaska.

[2] Network Security policy and objectives,
URL:http://publib.boulder.ibm.com/infocenter/iseries/securitypolco.htm

[3] Deep Inspection,
URL: http://www.ranum.com/security/computer_security/editorals/deepinspects /index.html

[4] Need of Network Security,
URL: http://www.indiastudychannel.com/resources/105777-Network-Security-Attackers-Hackers.aspx

[5] Packet filtering process,
URL: http://www.ibm.com/developerworks/linux/library/s-netip/

[6] Packet filtering using IPtables,
URL: http://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO-7.html

[7] eSoft, “Modern Network Security: The Migration to Deep Packet Inspection”, White Paper, 2006.

[8] John Peter Jesan, “Major threats to information security”, Graduate School of Computer Information Sciences Nova Southeastern University, 2005.

[9] McClure, Saumil &, Shreeraj, “Web Hacking : Attacks and Defense”, Boston: Pearson Education, Inc, 2003.

[10] Karen Scarfone, Peter Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”, Recommendations of the National Institute of Standards and Technology.

[11] Host Based IDS,
URL: http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html

[12] Wayne T Work “Intrusion Detection System What are They How do They Work”2003.

[13] Network- vs. Host-based Intrusion Detection “A Guide to Intrusion Detection Technology” 6600 Peachtree-Dunwoody Road 300 Embassy.

[14] David W Chadwick, “Network Firewall Technologies”, IS Institute, University of Salford, Salford, M5 4WT, England, 2008.

[15] Internet Firewall Tutorial, A White Paper July 2002.

[16] A. Hari, S. Suri, and G. M. Parulkar, “Detecting and resolving packet filter conflicts”, In Proc. of IEEE Infocom, pages 1203-1212, 2000.

[17] Frederick M. Avolio and Marcus J. Ranum, “A Toolkit and Methods for Internet Firewalls”, In Technical Summer Conference USENIX, Boston, Massachusetts, 2005.

[18] Christoph Ludwig Schuba, “On the modeling, design, and implementation of firewall technology”, Purdue University, 2008.

[19] Elizabeth D. Zwicky, Simon Cooper, D Brent Chapman,”Building Internet Firewalls”, 2-Ed, Oreilly, 2000.

[20] Ch14 : Linux Firewalls using IPtables,
URL:http://www.linuxhomenetworking.com/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables/

[21] Oskar Andreasson, “IPtables Tutorial”,
URL: http://www.frozentux.net/documents/iptables-tutorial/

[22] IPtables HowTo,
URL: https://help.ubuntu.com/community/IptablesHowTo

[23] IPtables Home,
URL:http://www.netfilter.org/

[24] IPtables Scripting,
URL:http://www.linuxdoc.org/

[25] IPtables command,
URL:http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/s1-iptables-options.html

[26] IPtables Parameters,
URL:http://www.centos.org/docs/2/rhl-rg-en-7.2/s1-iptables-options.html

[27] Wireshark,
URL: www.wireshark.org

[28] TCPdump,
URL: http://www.tcpdump.org/tcpdump_man.html

[29] IPP2P,
URL: http://www.ipp2p.org/docu_en.html

List of Publications

1. Bhisham Sharma, Ms. Sanmeet Bhatia, "Removing anomalies and Merging similar Firewall Rules", National Conference on Computing, Communication & Information Technology CCIT-2010, 28th-29th May, 2010 at Sant Longowal Institute of Engineering & Technology, Longowal.