

---

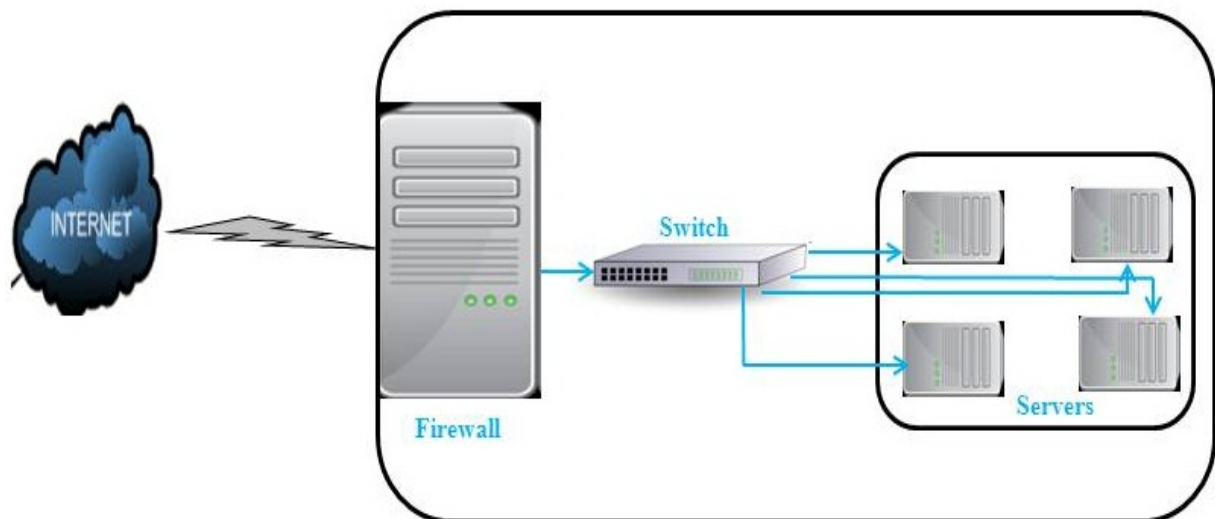
Technical report, IDE1202, February 2012

# Enhancing Network Security in Linux Environment

Master Thesis in Computer Network Engineering

By

Ali Mohammed, Sachin Sama and Majeed Mohammed



# Enhancing Network Security in Linux Environment

## Master Thesis in Computer Network Engineering

School of Information Science, Computer and Electrical Engineering  
Halmstad University  
Box 823, S-301 18 Halmstad, Sweden

February 2012

## **Preface**

First of all, we would like to express our sincere gratitude to our Supervisor Philip Heimer and Professor Tony Larsson for their supervision and assistance in the entire thesis work. We are also thankful to IDE department, Halmstad University for providing this opportunity to complete this thesis.

Ali Mohammed

Sachin Sama

Majeed Mohammed



## **Abstract**

Designing a secured network is the most important task in any enterprise or organization development. Securing a network mainly involves applying policies and procedures to protect different network devices from unauthorized access. Servers such as web servers, file servers, mail servers, etc., are the important devices in a network. Therefore, securing these servers is the first and foremost step followed in every security implementation mechanism. To implement this, it is very important to analyse and study the security mechanisms provided by the operating system. This makes it easier for security implementation in a network.

This thesis work demonstrates the tasks needed to enhance the network security in Linux environment. The various security modules existing in Linux makes it different from other operating systems. The security measures which are mainly needed to enhance the system security are documented as a baseline for practical implementation. After analysing the security measures for implementing network security, it is important to understand the role of network monitoring tools and Linux inbuilt log management in maintaining the security of a network. This is accomplished by presenting a detailed discussion on network monitoring tools and log management in Linux.

In order to test the network security, a network is designed using Linux systems by configuring different servers and application firewall for packet filtering. The security measures configured on each server to enhance its security are presented as part of the implementation. The results obtained while an unauthorized user accessing the servers from the external network are also documented along with attack information retrieved by different network monitoring tools and Linux inbuilt log messages.



# Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	MOTIVATION .....	1
1.2	GOAL.....	1
1.3	METHODOLOGY.....	2
1.4	REQUIRED RESOURCES .....	2
1.5	STRUCTURE OF THESIS .....	2
<b>2</b>	<b>BACKGROUND.....</b>	<b>3</b>
2.1	LINUX OVERVIEW .....	3
2.2	LINUX SECURITY .....	4
2.2.1	<i>Linux Security Module .....</i>	<i>4</i>
2.3	BASIC SECURITY MECHANISMS .....	5
2.4	FIREWALLS .....	8
2.4.1	<i>Circuit level Firewalls.....</i>	<i>8</i>
2.4.2	<i>Application level Firewalls .....</i>	<i>9</i>
2.4.3	<i>Packet filtering Firewalls.....</i>	<i>9</i>
2.5	NETWORK SERVICES OVERVIEW.....	11
2.5.1	<i>Apache web server .....</i>	<i>11</i>
2.5.2	<i>OpenSSL (Open Secured Socket Layer).....</i>	<i>11</i>
2.5.3	<i>Mail server .....</i>	<i>11</i>
2.5.4	<i>OpenSSH.....</i>	<i>11</i>
2.5.5	<i>File server .....</i>	<i>12</i>
2.5.6	<i>DNS Server.....</i>	<i>12</i>
2.6	NETWORK ATTACKS.....	12
2.6.1	<i>Reconnaissance Attack.....</i>	<i>13</i>
2.7	NETWORK MONITORING .....	14
2.7.1	<i>Linux Network Monitoring Tools .....</i>	<i>15</i>
2.7.2	<i>Ping – a basic Network Monitoring Program.....</i>	<i>15</i>
2.7.3	<i>Nmap (“Network Mapper”).....</i>	<i>16</i>
2.7.4	<i>Wireshark .....</i>	<i>17</i>
2.7.5	<i>IPtraf.....</i>	<i>18</i>
2.7.6	<i>Webmin.....</i>	<i>18</i>
2.8	LOG MANAGEMENT IN LINUX .....	19
<b>3</b>	<b>EXPERIMENT SETUP .....</b>	<b>21</b>

3.1	SECURITY MEASURES APPLIED IN SERVER CONFIGURATIONS.....	22
<b>4</b>	<b>RESULTS.....</b>	<b>26</b>
4.1	RESULTS IN SCENARIO-1 .....	26
4.2	RESULTS IN SCENARIO-2 .....	31
4.3	RESULTS COMPARISON.....	36
<b>5</b>	<b>CONCLUSION .....</b>	<b>37</b>
<b>6</b>	<b>REFERENCES .....</b>	<b>39</b>
	<b>APPENDIX.....</b>	<b>41</b>



## List of Figures

Figure 1. Circuit level gateway.....	9
Figure 2. Application level gateway .....	9
Figure 3. Packet filtering firewall.....	10
Figure 4. An Example of Ping Usage.....	16
Figure 5. An Example of Nmap Usage.....	16
Figure 6. Wireshark .....	17
Figure 7. IPtraf.....	18
Figure 8. Webmin.....	19
Figure 9. A sample showing output of folder /var/log .....	20
Figure 10. Block diagram for Implementation in real environment .....	21
Figure 11. Finding the IP address using Ping.....	26
Figure 12. The time taken by Nmap tool to access information of PC-1 before securing the system. ....	27
Figure 13. Attack information in PC-1 using IPtraf in Scenario-1.....	27
Figure 14. Attack information in PC-1 using wireshark in Scenario-1 .....	28
Figure 15. HTTP access from PC-4.....	28
Figure 16. FTP access from PC-4. ....	29
Figure 17. Output showing the information of access_log file in PC-1.....	29
Figure 18. Output showing the information of messages file in PC-1 .....	30
Figure 19. Output showing the information of server access in PC-1 using wireshark.....	31
Figure 20. The time taken by nmap tool to access information of PC-1 after securing the system. ....	32
Figure 21. Attack information in PC-1 using IPtraf in Scenario-2.....	32
Figure 22. Attack information in PC-1 using Wireshark in Scenario-2.....	33
Figure 23. HTTP access failure from PC-4. ....	33
Figure 24. FTP access failure from PC-4.....	34
Figure 25. Output showing information of error_log file in /var/log/httpd in PC-1.....	34
Figure 26. Output showing the information of secure file in /var/log/ in PC-1 .....	35
Figure 27. Output showing information of server access failure in PC-1 using wireshark.....	35

## List of Tables

<i>Table 1. Security Measures in Apache Web Servers .....</i>	<i>23</i>
<i>Table 2. Security Measures in FTP Server .....</i>	<i>23</i>
<i>Table 3. Security Measures in DNS Server.....</i>	<i>24</i>
<i>Table 4. Security Measures in SSH Server.....</i>	<i>25</i>
<i>Table 5. Security Measures in DNS Server .....</i>	<i>25</i>
<i>Table 6. Comparison of results in Scenario-1&amp;2.....</i>	<i>36</i>



## **1 Introduction**

Network security is an important task that must be seriously considered when designing a network. Network security is defined as the policies and procedures followed by a network administrator to protect the network devices from threats and simultaneously, the unauthorized users must be prevented from accessing the network [1]. As the numbers of attacks are increasing day by day, it is necessary to explore the information regarding new attacks and take appropriate steps to safeguard the network from malicious attacks. On the other hand, computer security is a main issue to be considered when using an operating system. Computer security mainly provides confidentiality and integrity to the data present in the system. It also authenticates users in accessing the system. Computer security ensures the security of one computer in a network whereas network security refers to the security of whole computer network. Generally, a network (also referred as a computer network) is defined as the interconnection of two or more nodes such as computers, servers, etc. A network can be as simple as two computers connected with a cable or it can be a large network that connects thousands of computers and other devices together [2].

### **1.1 Motivation**

Ensuring network security is often a complicated task employed by an organization. Every organization implements its own policies and procedures to protect its valuable resources from unauthorized access or damage. This process of securing the network and its resources can be achieved by enhancing the security of a designed network. Security enhancement can be accomplished by performing different tasks at various levels in the network design process. The security measures applied at each step in the network design process proves the security level of the designed network.

### **1.2 Goal**

The main goal of our thesis is to enhance the security in a network by performing various tasks in the network design process. The tasks performed in designing a secured network are:

- Configuring the servers by considering security measures.
- Configuring the firewall in such a way that unauthorized users do not enter or access the internal network information.

On the other hand, network security is tested by an unauthorized user trying to access the internal servers from the other network. The attack information is analysed by using network monitoring tools and log messages.

### **1.3 Limitation**

For enhancing the network security, we have focused on the security enhancement of different servers by considering security measures in the configuration and we have also configured the firewall for protecting the internal network. Apart from this, there are also some other tasks that enhance the network security such as user authorization techniques, data encryption techniques, etc., which are not covered in this thesis. Hence, this thesis implementation does not cover security enhancement related to users and communication links in the network.

### 1.4 Methodology

First, the Linux security module is discussed. The different Linux tools or utilities used in implementing security are explored. Secondly, a network topology is designed using Linux systems, which contain different server and firewall configurations. One of the systems is used as the attacker system for testing the network security. Results obtained when the attacker system trying to access the internal network along with attack analysis information are then documented. These results are considered in two scenarios. In **Scenario-1**, the results of testing the network before configuring the firewall and applying security measures in server configurations are explained. Whereas, in **Scenario-2** the network is tested after configuring the firewall and applying security measures in server configurations.

Basically, this thesis implementation consists of four systems (including attacker system) and a Switch. Due to the lack of equipment such as one system with two LAN cards, an approach of using VMware is adapted, in which only two systems are used, instead of four systems. Therefore, the results of two systems, i.e., the Firewall system (PC-1) and the Attacker system (PC-4), in the network are considered rather than four systems. PC-1 consists of all the servers such as DNS, HTTP, VSFTP, SSH, MySQL, etc., configured in a secured manner. An application firewall using Iptables is also configured in PC-1 itself. Hence, the desired results are obtained by designing the network in VMware with two systems, PC-1 and PC-4. However, the results are same for both physical and virtual resource utilization.

### 1.5 Required Resources

The proposed implementation basically needs four different PCs, in which one system needs two LAN cards and one switch to make communication between these systems. These four systems contain the Linux flavour, CentOS 5.7 operating system, kernel version 2.6.18-134, file system ext3(extended version of 3) and init version 2.85.

However, the virtual implementation requires VMware installed in only one PC. Two machines are to be configured with CentOS 5.7 in VMware and are connected as bridged connection.

### 1.6 Structure of Thesis

The work in our thesis is categorized as follows: chapter 2 explains the background work such as basic Linux security measures to safeguard a network which forms as a stepping-stone to build a secured network. A detailed report on network monitoring tools is also present in chapter 2.

Chapter 3 deals with the implementation part, which consists of different servers, tools and firewall configured in two systems and security measures applied in server configurations. Chapter 4 exhibits the results of the firewall configured system (PC-1) and attacker system (PC-4) explained in two scenarios and finally, our report ends with the conclusion and future work.

## 2 Background

### 2.1 Linux Overview

As the number of Internet users is increasing day by day, at the same time numbers of victims to the hackers are also increasing, this is because of using an unsecured operating system. An attacker mainly tries to attack the central part of an information system and usually the central part would be an operating system. The operating system in a computer decides the level of security [3]. If the operating system is unsecured, then it is easy for the attacker to modify any information in the computer. This leads to an evolution of a secured operating system such as Linux [4].

Some of the Linux functionalities are as follows.

**Flexibility:** Linux is flexible, as it supports high-performance server applications, desktop applications and embedded systems.

**Stability:** In the Linux system, if a new program or software is installed, it does not require to be rebooted periodically. Hence, it maintains the performance level of the system.

**Performance:** It does not degrade the performance level of the system even though it handles a large number of users simultaneously.

**Network friendliness:** Linux is a user-friendly operating system in terms of networking functionality such as; it can be easily configured as the server system or the client system depending on the requirement in the network.

**Security:** The Linux operating system is built with security features, as it provides the file access permission mechanism, which prevents the unauthorized users in gaining access to the files.

Linux also supports password security, file protection, virtual memory and multitasking. Networking for Linux was developed in such a way that it supports remote logins for large networks [5]. Linux machines are not expensive and are easy to maintain. Linux requires few hardware requirements such as 4Megabytes of memory and 80Megabytes of hard-disk space.

Linux is an open-source operating system, as it has the main beneficial features where users can modify the code. It is designed in such a way that it can run on different types of hardware. Linux also supports different types of servers such as Apache server and SSH server to run on it and it supports web browser like Mozilla Firefox [6]. Linux is one of the leading operating systems in servers, presently over 90% of supercomputers are running on some variants of Linux. Linux is used in a network because, it has a kernel programming interface, can support many users, can run many tasks, provides a secure hierarchical file system, is portable and has a large collection of useful utilities for system administration [7]. The basic-level security makes Linux more secure, such as stopping unnecessary services, deleting users and groups which are not in use, etc. Linux operating system supports in building firewalls, IPtables and squid proxy server. Linux IP-tables which are used between WAN and LAN, provide good security and data filter from WAN network [8].

#### **LINUX PROS:**

- It is free.
- It is portable to run on any hardware platform.
- Made to run continuously.

## Background

- Secure
- Scalable
- Support inter-process communication.
- The debug time is very short for Linux operating system and its applications [9].

## 2.2 Linux Security

For Internet users, security is one of the most important things. If sufficient security is not provided to the gateway, then malicious attacks may attack the gateway and degrade its performance. Usually network attackers will try to get information like file data, services, etc. The network attacks are of two types; one of them is, the attacker executes a malicious process and then tries to get control over the computer, such as worm attack and the other is, it first tries to get the system under control and then make the system unavailable for the service. To avoid these types of attacks, we need to improve the network safety-defence mechanism of an operating system. Linux is the best choice for its open source and secure environment. Linux has Linux kernel, which provides Socket API to accomplish all the network actions. These Socket APIs, if handled properly, help us to keep malicious attacks away [10].

In a network security policy, the main points to be considered are firewall, Intrusion Detection System (IDS) and so on, these show the level of security for the internal server which contains important information. The Linux security has functions like authentication, which identifies the users and log management, which records the network activities. Apart from these functions, it contains the Linux kernel, which enhances the network security. The Linux kernel is high-capacity software, which performs the functions like process management, file system, network management and memory management. There are few requirements for Linux security, they are:

- Managing user authentication and accounts.
- Access control on file and directory.
- Process management.
- Network access control.
- Hacking prevention functions.
- Functioning of self-protection.
- Installation and performance [11].

### 2.2.1 Linux Security Module

In Linux systems, the Linux Security Module (LSM) allows the kernel to support a variety of computer security models instead of using a single security model. In the present main-stream operating system, the access control mechanisms are not sufficient for providing strong system security. The enhanced security control mechanisms were even not good enough for the main-stream operating system to accept it. It could satisfy only a normal user but not for a large number of users.

In LSM, the different access control mechanisms are carried out as loadable kernel modules for the main-stream Linux kernel which was developed for lightweight, general purpose access control framework and the LSM framework was adopted by several existing enhanced

## Background

access control implementations like Security Enhanced Linux (SELinux), Domain and Type Enforcement (DTE) and the Linux Capabilities [12].

### **SELinux:**

Security Enhanced Linux (SELinux) is one of the security modules, which are included in few of the Linux distributions. SELinux enhances the security in the Linux distribution [13].

SELinux is developed by NSA (National Security Agency), and it allows the users and administrators to control the access of data or application. It controls the access issues such as, which users can access the applications or data and to what extent the users have authorization over the data, such as (-rwxr -xr -x ) read, write and execute permissions. SELinux also provides networking resources and inter-process communications.

SELinux has three basic modes of operation:

- Enforcing
- Permissive
- Disabled

### **Enforcing Mode:**

On a system, SELinux security policy is enabled and enforced, this mode is the default mode.

### **Permissive Mode:**

In permissive mode SELinux is enabled but not enforcing the security policies; it does not rectify the problems, but it warns about the problems.

### **Disabled Mode:**

SE Linux is turned off [14].

## **2.3 Basic Security Measures**

Below are some of the basic ways to secure a Linux system.

**Removing unnecessary software packages (RPMs):** Removing unnecessary packages are very important for security management. To achieve this, the system administrator should have a full knowledge of the system, such as which applications are running and which applications are mostly used by the users, etc. If not, in critical situations, it would be difficult to understand which packages should be removed and which packages are to be secured. Therefore, it is very important for the administrator to have a clear understanding of the packages available in the system before removing the unnecessary packages. This increases the system performance and also there will be fewer software packages for updating and maintaining [15].

Before removing any packages from the operating system, the best thing is to know how many packages are installed. This can be accomplished by issuing following commands.

```
# rpm -qa * (will provide all the package names)
```

```
# rpm -qi package name (to know briefly about that package)
```

```
# rpm -e - -test package name (removing package and reporting dependency)
```

**Patching Linux systems:** Patching is one of the most important tasks of the Linux system administration. Patching refers to installation of latest updates of the operating system or application. The latest patches provide improved security to the applications. A separate



## Background

security log should be maintained in detail about the Linux security notices received, patches explored, assessed and applied, etc. Patching also rectifies security loopholes in the system [15].

There are different methods available for updating packages. By using the Internet we can set an auto update option, or we can download and install updates manually. If you want to update offline, then have to create yum repository in:

```
/etc/etc/yum.repos.d/reponame.repo
```

```
# yum list updates (provides update information)
```

```
# yum update (update complete OS with kernel)
```

```
# yum list installed (provides installed packages information)
```

```
#yum update package-name (to update a particular package)
```

```
#yum list all (displays available packages)
```

```
# yum groupupdate "package-name" (update all the group packages) [15].
```

**Detecting listening network ports:** For maintaining a secured system, it is important to detect and close network ports that are not in use. For doing this, it is important to know the type of applications running on each port. This helps the administrator to detect the unnecessary application running on a specific port and stop those listening ports [15].

Information about listening port numbers can be gathered by issuing the following commands:

```
# netstat -tunlp (lists all the services running along with port numbers)
```

```
# nmap -sTU destination ipaddress (used to scan a system remotely).
```

```
# lsof -i -n | grep 'COMMAND|LISTEN|UDP' ( provides all the TCP and UDP listing port numbers) [15].
```

**Disabling unnecessary system start-up services:** It is good practice to stop unnecessary services during system start-up. For this, it is important to have knowledge about the booting process of the Linux operating system [15].

It is a better idea to stop all unnecessary applications at start-up.

```
# chkconfig --list | grep on (provides a list of services running at start-up)
```

Related script file is **/etc/init.d**

If we want to stop some services like VSFTP at start-up time we can use this command

```
# chkconfig vsftpd stop or #/etc/init.d/vsftpd stop [15].
```

**Restricting system access from servers and networks:** Generally, a firewall is used to protect internal servers and networks from outside networks. Sometimes, these servers are also in need of protection from internal network access. This can be accomplished by using TCP wrappers. Xinetd is one of the services, which has built-in TCP wrappers. The access control mechanism in TCP wrappers is implemented by using two files: /etc/host.allow, /etc/host.deny [15].

**Securing remote login:** Most operating systems support remote login by using telnet, rlogin and rsh. By default, these services are not secure, because these services are vulnerable to attack such as eavesdropping. So, instead of using these unsecured services for remote login,

## Background

it is the best practice to use SSH. Moreover, it does not allow the root user to login from outside service and SSH version 2 is more secure than SSH-1 because more security threats are rectified in SSH 2 version [15].

**Checking accounts:** It is very important that all the system's unused logins should be locked. This can be performed by using some utilities in the Linux operating system [15].

Unnecessary user accounts are disabled by using some of the following utilities.

```
# find / -path /proc -prune -o -user username -ls ( shows unused accounts)
```

```
# userdel -r username (to delete a user)
```

```
# usermod -L username (to lock a user)
```

```
# usermod -u username (to unlock a user)
```

```
# id username (to get the user group info)
```

```
# change -E yyyy-mm-dd username ( make expiration of a user) [15].
```

**Enforcing stronger passwords:** Assigning most difficult password is also one of the best practices for security management, the password should contain alpha numerical with special characters, for example, (some#7((U))@!L). The password should not be a dictionary word or a guessable word because, it is cracked easily by the attacker using simple password cracking tools like 'John the ripper' etc. Then, this leads to a big disaster [15].

**Lock user accounts:** It is always better to lock a user account after making two or more login fails.

**Restricting direct logins:** It is good to know which user is using which system or shared account on an audited production system. If the password is known to more than one person, then the direct logins for all systems and shared accounts must be restricted. If a user wants to switch to a system or a shared account, then he has to use his own account details for the direct login [15].

**Preventing accidental Denial of Service:** In Linux, for a user or a group of users the resources can be set to a limit. This is very helpful in situations such as, if a bug in a program accidentally starts using too many resources and making the machine slow down. There are many situations in which we use incorrect settings and that results in allowing the programs to use more resources than needed. This makes the server unavailable to new connections and local logins [15].

**Security level of User accounts:** In Linux, each user can have different levels of permissions, while creating a new user ID the system administrator can grant specific permissions to the account so that he/she cannot access other user group information. It is also good to block telnet privileges for others, giving them only the FTP permissions, which can keep them away from other accounts [15].

**Control network user's access to a system:** By installing the TCP wrappers, limiting of IP address can be achieved; this controls the IP address of services like FTP, telnet, rsh, finger and so on. So, by this we can allow only few machines to operate the server within the company. Limiting the super user password usage is also important. In this, only the system administrator should know the super user password and he/she has to change the password frequently and should not allow telnet to access the root. He can use 'sudo' to authorize some actions. And while implementing system security, there are several fundamental concepts that can make a system more secure [6]. Patch management (up-to-date) and disabling

## Background

unnecessary services are good, but overall, limiting permissions, security policy usage, system auditing and log management will make a system more secure [8][16].

By utilization the above safety rules, we can improve the Linux system's security to be improved and also the user must use the firewall to defend against hackers, along with fundamental security policies to guarantee the system security [17].

**Kernel tuning:** The important component of the Linux operating system is the kernel. It provides communication between application software and hardware components of the computer. In Linux, these kernel parameters can be tuned to protect the internal servers from attacks. This enhances the security of Linux servers in a network. The main script file of kernel is `/etc/sysctl.conf`. The desired parameters, which are to be tuned, are entered in this file and the system is reloaded.

For example, if we want to protect the system from IP spoofing, the source address verification should be enabled. This can be done by adding the following line to the file `/etc/sysctl.conf` [15].

```
net.ipv4.conf.all.rp_filter = 1
```

## 2.4 Firewalls

A firewall is one of the most widely used solutions for the Internet world. All traffic inside to outside and vice versa, must pass through the firewall. Different types of firewalls have different types of rules and security policies. The authorized traffic will be sent based only on local policies. The firewall itself is protected, i.e.; it uses a trusted hardware and operating system. Generally, firewalls are of three types.

- Circuit level firewalls
- Application level firewalls
- Packet filtering firewalls

### 2.4.1 Circuit level Firewalls

For certain applications the circuit level gateway can be either a stand-alone system or a specialized function performed by an application level gateway. End-to-End TCP connections in a circuit level gateway are not permitted but instead of that, the gateway creates two TCP connections, one between the gateway and the TCP user on an inner host and the other between the gateway and the TCP user on the outside host. When the TCP connection is established the gateway exchanges the TCP segments without examining the content. The security functions only determine which connections are to be allowed [18]. The behaviour of the Circuit level firewall for establishing the connection between the inner and outer host is shown in the Figure 1.

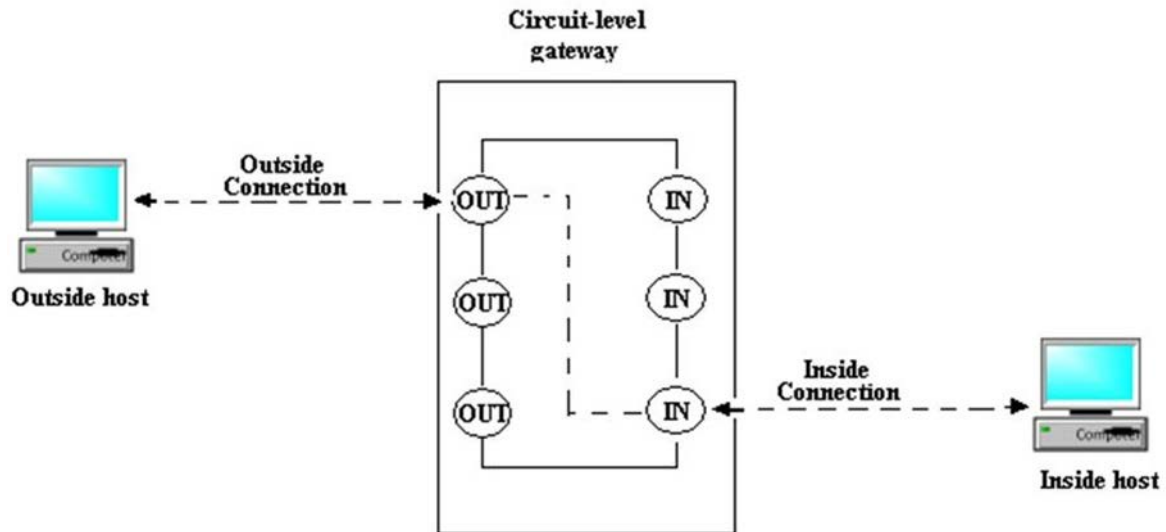


Figure 1. Circuit level gateway

#### 2.4.2 Application level Firewalls

An Application level gateway which is also called a proxy server, acts as a relay of application-level traffic. A user can contact the gateway by using a TCP/IP application and then the gateway asks about the remote host which is to be accessed. Then in response the user must give a valid user ID and authentication details, then the gateway contacts the application on the remote host and exchange the TCP segments application data between the two end points. However, to perform these things the gateway must implement the proxy code. We can configure the gateway to support only the particular features of an application [18]. The behaviour of Application level firewall in connecting the two ends is shown in Figure 2.

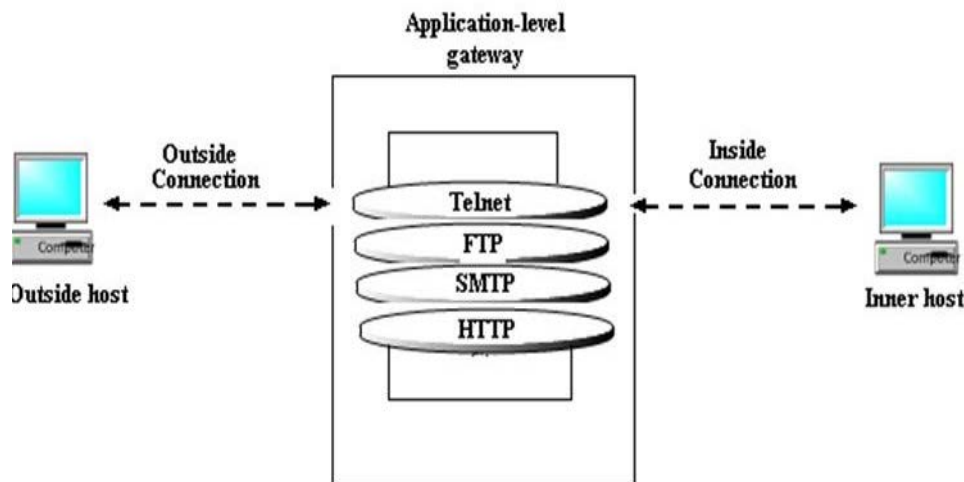


Figure 2. Application level gateway

#### 2.4.3 Packet filtering Firewalls

The packet filtering is done based on the set of rules configured on a packet filter router. The packet is forwarded or discarded based on the configurations done. There are two default policies involved in forwarding or discarding the packet, they are:

## Background

- **Default:** Discard (Which doesn't match the set of rules)
- **Default:** Forward (Which matches with at least any one of the rules)

If a packet matches with at least any one of the rules, then one of the default actions takes place i.e., it is forwarded, and if it doesn't match with any of the set of rules, then the other default action takes place i.e., discards the packet. The block diagram of a Packet filtering router is shown in Figure 3.

The packet filtering is done based on information in the network packet.

- **Source IP address:** The IP address from where the packet is originated.
- **Destination IP address:** The IP address to which it wishes to send.
- **Transport-level address of source and destination:** The applications like SNMP or telnet which are defined by the transport level port (TCP or UDP)
- **IP protocol field:** Transport protocol is defined.
- **Interface:** From which interface of a router the packet is originated and to which interface of a router the packet is destined to be sent. [18]

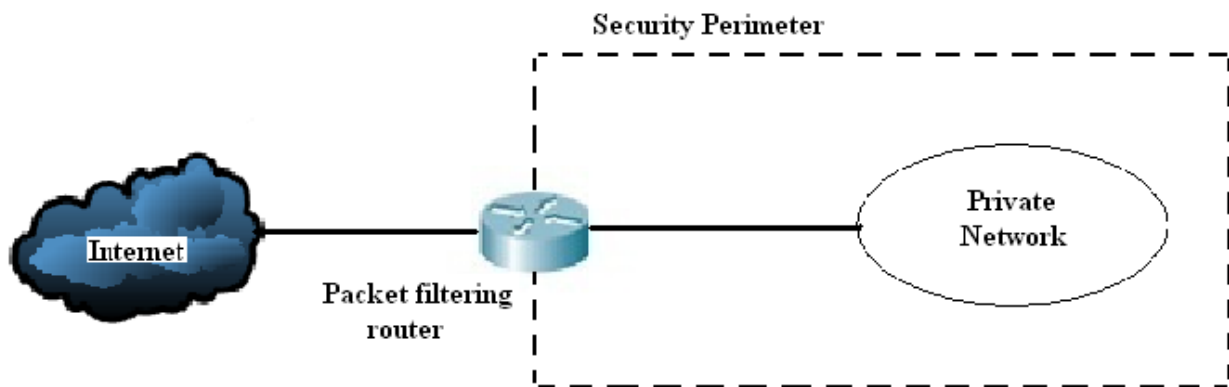


Figure 3. Packet filtering firewall

### IPtables:

IPtables can also be used for packet filtering. Generally, in Linux IPtables are popularly known as IPchains. The Netfilter organization has introduced a new version of IPchains with a new name called IPtables, which is more secure than IP chains. There are some advantages given below:

- Better Network address translation.
- IPtables has a rare limiting feature which helps in blocking few types of DOS.
- For a detailed report, firewall and system logging can be integrated.
- Transparent integration is well supported by proxy programs like Squid.
- Kernel modules are designed for improved speed and reliability with the integration of Linux Kernel, which has the capabilities of loading IPtables.
- A firewall always has an eye on each connection passing through it, and in some cases to know about the preceding actions of a particular protocol, a firewall would look at the content.

## Background

- Based on MAC address and the values of the flags in the TCP header the packet filtering is done.

## 2.5 Network Services overview

The servers which are commonly found in a network are discussed in the following sections.

### 2.5.1 *Apache web server*

Apache web server is one of the widely used web servers in the world, as it possesses multi-threading concepts. In Apache server, both HTTP and HTTPS services are available. HTTP protocol is designed to deliver the communication between the clients and servers. By default, it runs on the port number 80. HTTP is used to establish normal connections.

HTTPS runs by default on port number 443 and it establishes secured connection. When establishing the HTTPS connection, the server responds to the client with a list of encryption techniques. In return, the client prefers the better connection mechanism. Therefore, the authentication of servers and clients is verified by exchanging the certificates and encrypted information in order to ensure that both use the same keys. HTTPS is most widely used in login pages of banks and corporate companies.

### 2.5.2 *OpenSSL (Open Secured Socket Layer)*

SSL is an open-source tool kit which is implemented by the two layered protocols. They are: SSL v2/v3 and transport layer protocol. It uses the strong cryptography library. The current version of openssl is 1.0.0e, that includes bugs and security fixes. OpenSSL supports many numbers of cryptographic algorithms and protocols.

SSL provides better security for web services. OpenSSL records the confidentiality and integrity for the SSL connections and also provide better security features to the higher layered protocols. For HTTP, it provides the transfer service for web client/server interaction that can operate on top of SSL [19].

There are also other three higher level protocols, which have a key role in the management of SSL exchange, they are:

- Handshake protocols.
- Change cipher protocol.
- Alert protocols [20].

### 2.5.3 *Mail server*

Sendmail server is one of the best mail servers used in the majority of real environments. By default, Sendmail server can only send mails; it can't receive any mail. It is very important that receiving mail should be properly scanned and checked. Sendmail supports a number of mail transfers and delivery methods such as SMTP, which is used for email transport over the Internet.

### 2.5.4 *OpenSSH*

OpenSSH is a free version of SSH, and it provides the encrypted communication over the Internet. The tools such as telnet and rsh are insecure because they transmit passwords in clear text format. OpenSSH provides a better security service during the transfer of files.

## Background

For client connections, the SSHD (a component of SSH server) listens continuously from any client tools. OpenSSH uses different types of authentication methods, such as regular passwords and public keys [21].

Generally, telnet, rsh, and rlog are not safe remote applications because these are prone to eavesdropping. For this reason, most companies will prefer SSH for remote logins.

### 2.5.5 File server

FTP is a widely used protocol, which is used to download files from the Internet. Linux uses VSFTP protocol, which is simple, fast and secure. FTP server is used to transfer and copy files from servers to the Internet and vice-versa. FTP operates in two modes of connection channels, they are FTP control channel and data channel.

FTP control channel runs on TCP port number 21, and all the commands sent to the FTP server and responses are handled by the control connection channel whereas an FTP data channel runs on TCP port number 20, and is used to transfer the data between servers and clients.

### 2.5.6 DNS Server

DNS stands for Domain Name System and it is a hierarchical distributed database. It translates domain names into IP addresses and vice-versa. As the domain names are alphabetic, they are easier to remember than IP addresses, which contain numbers. However, any public or private networks are based on IP addresses but not domain names. Each time when we use a domain name, it is the DNS server which translates the name into the corresponding IP address. For example, the domain name [www.example.com](http://www.example.com) might be translated to 192.168.40.132. Every network contains one or more DNS server. If one server fails to translate a particular domain name, it is handled by another server and so on. In brief, DNS serves as the phone book for the Internet by mapping the directory of domain names with IP addresses and vice-versa. Therefore, securing DNS server is an important part of securing a network.

#### **DNSSEC:**

DNSSEC is originated to resolve security issues in the DNS server. It is the extension of DNS which provides security features to DNS clients in terms of authentication of DNS data, data integrity and authenticated denial of existence, through the validation of a digital signature measure [22].

## **2.6 Network Attacks**

Network attack can be defined as an operation which destroys the services available in the network.

Day by day, attackers are generating different types of attacks, which cause serious problems to any network. So, in order to get rid of problems caused by attacks, a network administrator must be updated with the latest security technologies and should be able to defend and mitigate any network attacks generated within the network [23].

#### **Need For Network security:**

Security is the basic need of every network, while designing the network, strong security policies should be implemented.

## Background

In earlier days, attackers were highly experienced programmers, who had detailed knowledge about computer communication, but today, anyone can become a hacker just by downloading any hacking tools from the Internet. As the numbers of hackers is increasing, this leads to an increase in network security policies.

### 2.6.1 Reconnaissance Attack

A reconnaissance attack is a method of collecting the information of a network system and services, to exploit the vulnerabilities within the network. The attacker mainly targets well reputed organizations by pinging their domain names to get the desired information. Reconnaissance attack techniques are classified into two types

- Passive reconnaissance attacks
- Active reconnaissance attacks

Passive reconnaissance attack is the process of gathering the network information by direct or indirect methods. The indirect passive method collects the information through the databases and Internet domain registration services, whereas the direct passive method gathers the information directly by using Dig, NS LOOKUP, and DNS traversal tools.

Active reconnaissance attack is defined as the process of gathering the information with direct contact to the target network. Some of the common tools of active reconnaissance attack are Nmap, Xprobe and hping2 etc. [24].

Some of the reconnaissance attacks are:

- Port scans
- Ping sweeps
- Internet information queries, etc.

#### **Port scan:**

A port scan is one of the reconnaissance attacks, where the attacker tries to scan the target host to know the services provided. Before attacking the network, the attacker performs a port scan to identify the available services.

The attacker normally uses different types of scanning tools to scan the network; some of the tools are used to scan only specified ports while others are used to scan only the TCP ports. Among the scanning tools, TCP port scanning is considered as the effective scanning as it identifies the target machine vulnerabilities, which can be easily attacked by the attacker. Whenever the attacker scans the remote computers, there is a possibility of getting three types of responses:

- Open port.
- Closed port.
- Stealth port.

In case of open port mode, the ports are open, and the attacker can easily get connected with the open ports. These ports are in standby mode and wait for the incoming connections.

In closed port mode, the ports are in closed state and the attacker cannot connect to the target host as it has a better security features enabled.



## Background

In stealth port mode, an attacker may or may not receive any kind of information from the target machine. It mainly depends on the security configured on the target host.

The main objectives of a port scanning are to:

- Find the active target host on the network.
- Check whether TCP/UDP ports are active in the network.
- Usage of operating system, applications and software on the target host.

### **Ping sweep:**

Ping sweep is the oldest and slowest mechanism to scan the network. In ping sweep mechanism, ICMP echo requests are sent to multiple hosts in the network and if the given hosts are active then they send the ICMP echo reply notification messages to the attacker. A Ping request also provides the information such as time taken to retrieve the notification messages and packet loss information.

The main objective of ping sweep is to find the active hosts in the network. Tools which are used in the ping sweep are fping, gping and Nmap. Single ping scans only single host whereas fping scans a list of IP addresses from the file so that the attacker does not have to enter the addresses each time, it pings one host after another host without waiting .

### **Internet Information Queries:**

Internet information queries are used by the attackers to gather the information of the target host from the public domains, such as the Internet. The attacker uses Internet tools such as WHOIS, nslookup, Dig, to find the information.

A WHOIS tool is used to make a query in the Internet and is used to retrieve the information such as IP address and its owner information [25]. An attacker also uses the DNS query mechanism to retrieve information from the DNS server. For example, an attacker sends a query such as domain name to the DNS server and then the DNS server searches the data of the particular domain and gives the reply with IP address and its owner information [26].

## **2.7 Network monitoring**

Network Monitoring refers to a practice of monitoring the traffic over a computer network using different network management tools. Availability and performance of the network services and hosts are ensured by network monitoring tools or systems. Moreover, they are also helpful in providing information about traffic transiting within the network [27].

### **Features of Network Monitoring Systems or Tools:**

- Detection and reporting the failure of devices or connection in the network can be solely done by network monitoring systems or tools.
- They normally measure the CPU utilization, network utilization; amount of traffic transferring in the network, type of request accepted and replied by the hosts in the network.
- Network monitoring tools notify a network administrator or other hosts in the network (such as a management server, an email server or a phone number) through the messaging system.
- In addition, network monitoring also provides essential data to other network management processes such as network security, accounting and optimization.

## Background

Currently, there are many network monitoring tools available, which differ from one another depending on their specific purpose. For instance, some network monitoring tools focus on packet inspection rather than traffic flow, whereas the others focus on traffic flow along with inspection of packets. However, it is the duty of a network administrator to choose which network monitoring tool best suits for a particular purpose [37]. Thus, monitoring traffic in a network is as important as protection of data and devices in that particular network [28].

### 2.7.1 *Linux Network Monitoring Tools*

Linux is one of the best platforms to learn network troubleshooting techniques. It offers a number of inbuilt command line tools to detect and diagnose a network problem. In addition, there are several open-source network monitoring tools both with graphical and command line interfaces, which help to visualize and analyse the network traffic. But, as per our earlier discussion, it depends on the network administrator to choose the best monitoring tool that suits the particular purpose from the pool of tools. Thus, after serious research on network monitoring tools, we have used some of the tools in our implementation (discussed in the following paragraphs) to monitor the network traffic, which includes both graphical and command line tools. Moreover, a brief discussion about a web-based interface tool called *webmin*, for system administration is also documented.

Given below are some of the monitoring tools we have used in our thesis implementation:

- Nmap
- Wireshark
- IPtraf

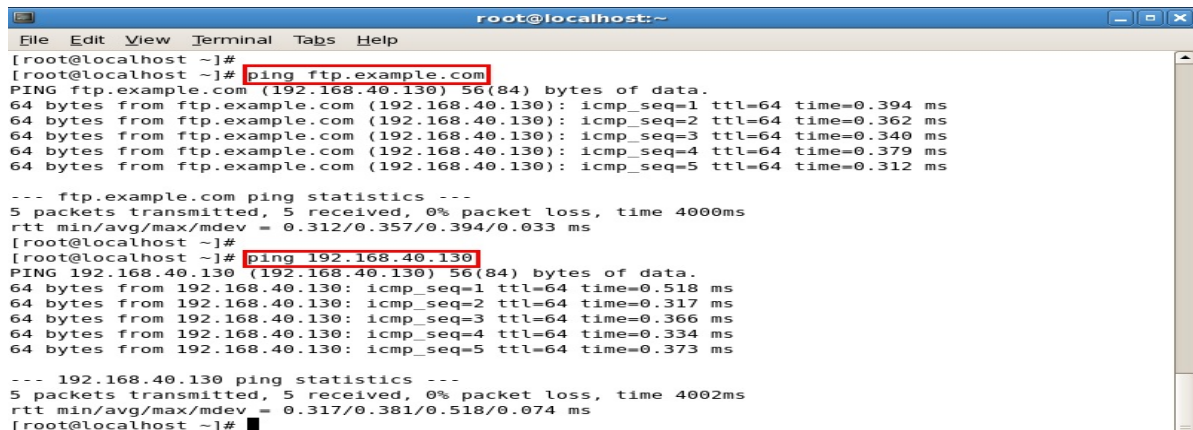
### 2.7.2 *Ping – a basic Network Monitoring Program*

Ping is an in-built software tool available in most of the network devices. It is one of the basic network monitoring programs used to verify whether the connection between two network devices/hosts in a network is active or not. The ping application sends ICMP ECHO\_REQUEST packets to the target host. If the host is active, then it gives a response to the request sent by the ping. The command also shows the time taken by the target to respond and some additional information as ping statistics. Thus, the information provided by the ping application forms the basic step in the network troubleshooting process.

The general syntax of ping command is: `ping hostname`

Example of ping usage: `ping ftp.example.com` or `ping 192.168.40.130`

## Background



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ping ftp.example.com  
[root@localhost ~]# ping 192.168.40.130  
PING ftp.example.com (192.168.40.130) 56(84) bytes of data:  
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=1 ttl=64 time=0.394 ms  
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=2 ttl=64 time=0.362 ms  
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=3 ttl=64 time=0.340 ms  
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=4 ttl=64 time=0.379 ms  
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=5 ttl=64 time=0.312 ms  
  
--- ftp.example.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4000ms  
rtt min/avg/max/mdev = 0.312/0.357/0.394/0.033 ms  
[root@localhost ~]#  
[root@localhost ~]# ping 192.168.40.130  
PING 192.168.40.130 (192.168.40.130) 56(84) bytes of data:  
64 bytes from 192.168.40.130: icmp_seq=1 ttl=64 time=0.518 ms  
64 bytes from 192.168.40.130: icmp_seq=2 ttl=64 time=0.317 ms  
64 bytes from 192.168.40.130: icmp_seq=3 ttl=64 time=0.366 ms  
64 bytes from 192.168.40.130: icmp_seq=4 ttl=64 time=0.334 ms  
64 bytes from 192.168.40.130: icmp_seq=5 ttl=64 time=0.373 ms  
  
--- 192.168.40.130 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4002ms  
rtt min/avg/max/mdev = 0.317/0.381/0.518/0.074 ms  
[root@localhost ~]#
```

Figure 4. An Example of Ping Usage

### 2.7.3 Nmap (“Network Mapper”)

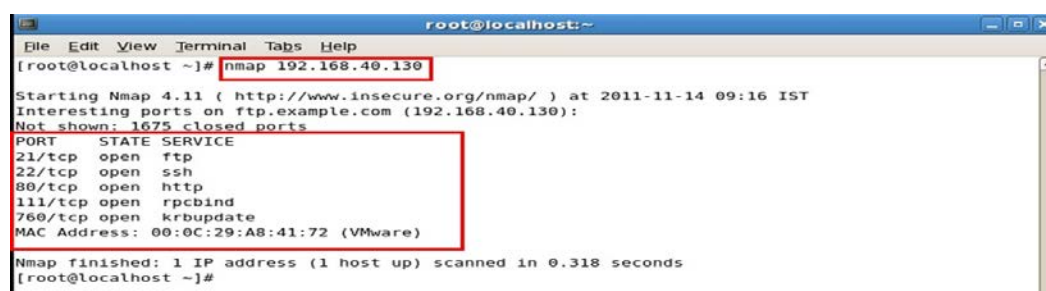
Nmap is an open-source tool, widely used in exploring the network or security auditing. It is also used as a hacking tool to gather the information of the target machine. It is mainly used by the network administrator for performing tasks such as:

- Network inventory management.
- Managing service upgrade tasks.
- Monitoring host or service start-up time [29].

It sends raw IP packets to the target network/system to gather the information for any purpose. The information gathered by the nmap about the target network includes:

- Hosts available in the network.
- Services offered by the hosts (along with application name and version).
- The type of operating system running on the target systems (including name and version).
- Type of firewalls or packet filters in use, and a lot of other useful information.

Nmap is compatible to use on all computer operating systems such as Windows, Linux, and Mac OS. It is normally used as command line executable file but GUI also exists. Basically, nmap was designed to scan the large networks but works fine to scan a single host [29]. Before moving forward on nmap description, one important point to be noted is, this nmap is best used by hackers as a hacking tool rather than as a network monitoring tool by an administrator. Figure 5 is a sample output of information gathered by the nmap utility in our implementation.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# nmap 192.168.40.130  
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2011-11-14 09:16 IST  
Interesting ports on ftp.example.com (192.168.40.130):  
Not shown: 1675 closed ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
111/tcp   open  rpcbind  
760/tcp   open  krbupdate  
MAC Address: 00:0C:29:A8:41:72 (VMware)  
  
Nmap finished: 1 IP address (1 host up) scanned in 0.318 seconds  
[root@localhost ~]#
```

Figure 5. An Example of Nmap Usage

## Background

In the Figure 5, nmap is used to gather information about one of the hosts in our network. It gives the information about services which are offered by that particular host along with TCP port numbers used by the services. It gives the MAC address of the host as 00:0C:29:A8:41:72 and the type of machine as VMware. It gives information in a short amount of time if the target system is not securely configured, otherwise it takes longer time.

### 2.7.4 Wireshark

Wireshark is a free and open source network packet analyser which was originally named as **Ethereal**. The main task of this network packet analyser is to capture network packets and display that packet data in detail. It can be used by many people in different situations. Some of the examples of situations where wireshark is helpful are:

- For troubleshooting network problems -used by the network administrators.
- For examining security problems -used by the network security engineers.
- For debugging protocol implementation –used by the developers.
- For analysis and education.

In order to get the entire traffic information of a network interface, Wireshark allows the users to put the interface in promiscuous mode. Wireshark provides the information along with different tools to filter and display, based on a number of criteria such as source or destination address, protocol, error status, etc. One of the most important points to be considered is, Wireshark will not warn when someone is doing any harmful things in the network, which he/she is not supposed to do. It will just figure out the things going-on in the network. Thus, Wireshark is not an intrusion-detection system; it is just a network packet analyser [30].

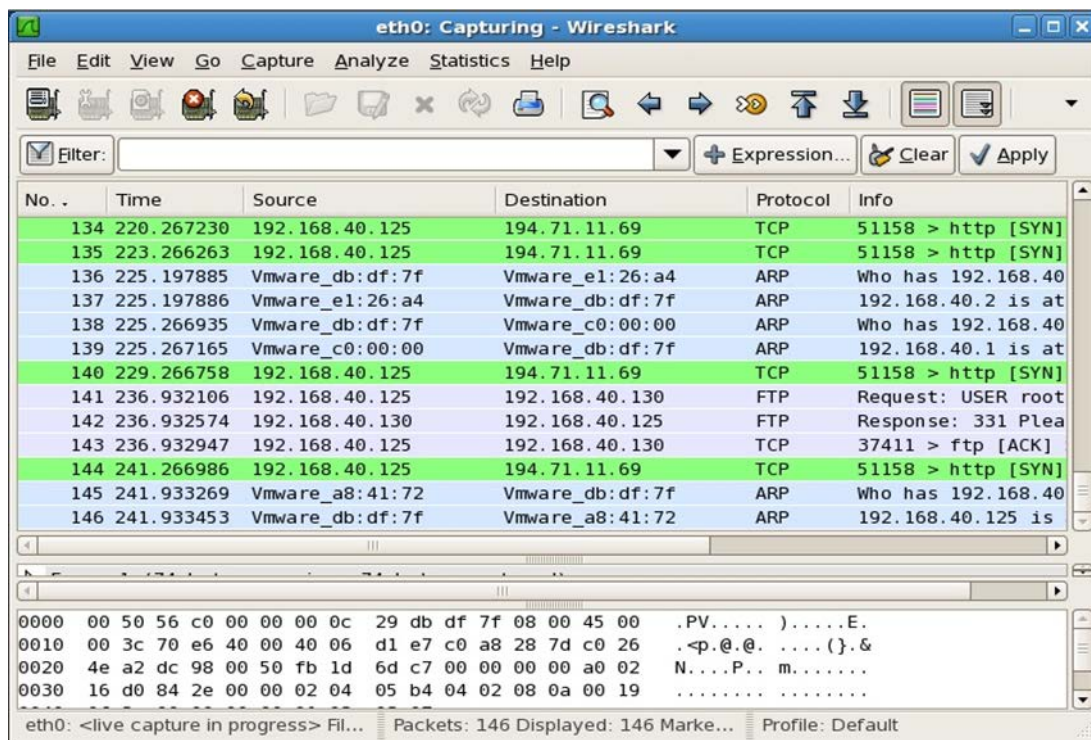


Figure 6. Wireshark

## Background

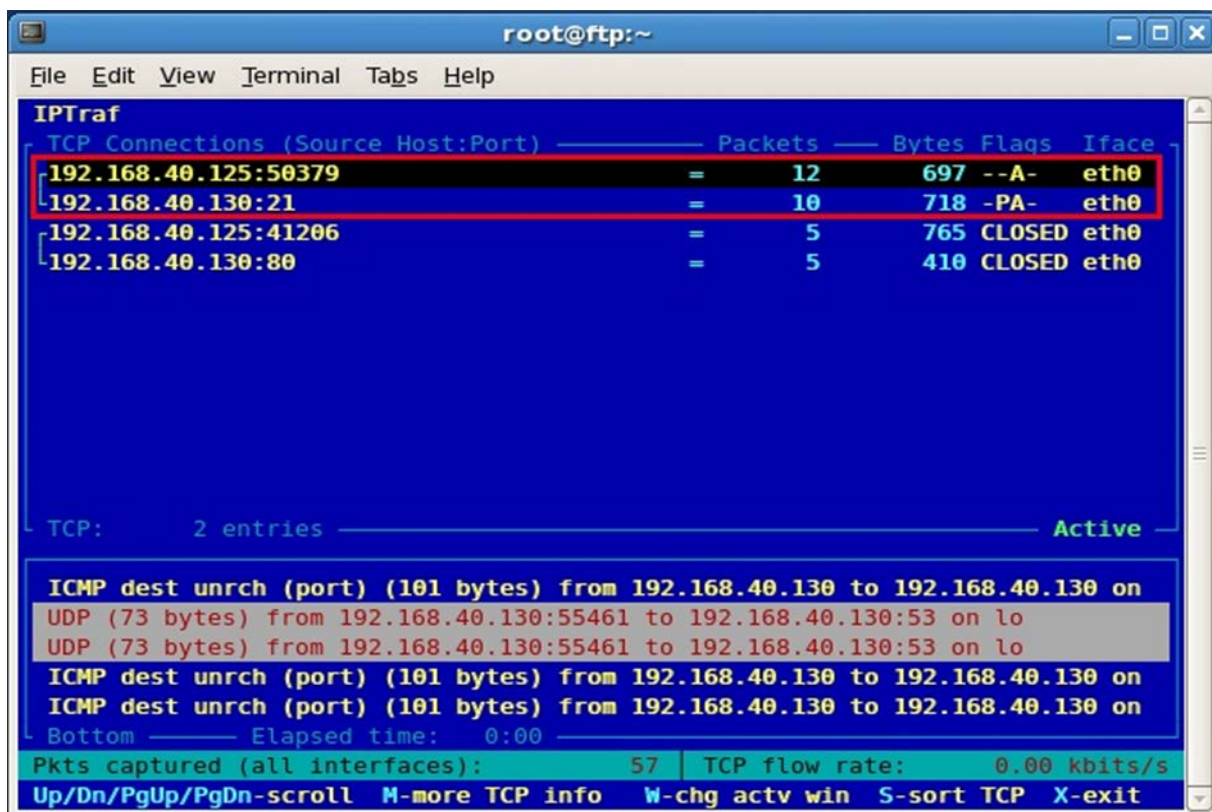
### 2.7.5 IPtraf

In Wireshark, we get a lot of detailed information about the traffic transiting in the network, and in order to understand every single point of information a lot of time can be wasted. Hence, to make the things more specific we used IPtraf. It is a reliable console based network monitoring utility for Linux. It gathers following information:

- TCP connection packet and byte counts.
- Interface statistics and activity indicators.
- TCP/UDP traffic breakdowns.
- LAN station packet and byte counts.

It is one of the best network monitoring utilities for Linux, which keeps an eye on the exchange of IP packets to and from the machine. The best features of IPtraf make it more specific to figure out the information of IP traffic passing over the network. Thus, the information gathered by the IPtraf can be useful in various situations such as:

- Making decisions in an organization.
- Troubleshooting networks.
- Tracking of various IP hosts [31].



The screenshot shows the IPtraf terminal window with a blue background. The title bar reads 'root@ftp:~'. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The main display area shows a table of TCP connections with columns for Source Host:Port, Packets, Bytes, Flags, and Iface. The first two rows are highlighted with a red box. Below the table, there is a section for ICMP and UDP traffic. At the bottom, there is a status bar showing 'Pkts captured (all interfaces): 57' and 'TCP flow rate: 0.00 kbits/s'. The status bar also includes navigation keys: 'Up/Dn/PgUp/PgDn-scroll', 'M-more TCP info', 'W-chg actv win', 'S-sort TCP', and 'X-exit'.

TCP Connections (Source Host:Port)	Packets	Bytes	Flags	Iface
192.168.40.125:50379	= 12	697	--A-	eth0
192.168.40.130:21	= 10	718	-PA-	eth0
192.168.40.125:41206	= 5	765	CLOSED	eth0
192.168.40.130:80	= 5	410	CLOSED	eth0

TCP: 2 entries Active

ICMP dest unrch (port) (101 bytes) from 192.168.40.130 to 192.168.40.130 on  
UDP (73 bytes) from 192.168.40.130:55461 to 192.168.40.130:53 on lo  
UDP (73 bytes) from 192.168.40.130:55461 to 192.168.40.130:53 on lo  
ICMP dest unrch (port) (101 bytes) from 192.168.40.130 to 192.168.40.130 on  
ICMP dest unrch (port) (101 bytes) from 192.168.40.130 to 192.168.40.130 on

Bottom Elapsed time: 0:00

Pkts captured (all interfaces): 57 TCP flow rate: 0.00 kbits/s

Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit

Figure 7. IPtraf

### 2.7.6 Webmin

Webmin is a web-based system administration tool for Linux (recent versions also support Windows). It is a Graphical User Interface tool, through which one can take complete control of servers using a web browser. That is, it runs on any modern web browser through which we



## Background

can remotely manage our system. With Webmin, it is possible to control/configure some of the following functions.

- Operating system internals like setup user accounts, disk quotas, etc.
- Configure services and their configuration files [32].
- Fine-tuning control over the system security and resources.
- Modify and control server configurations such as Apache, DNS, FTP, etc.

By default, Webmin uses TCP port 10000 for communication. It is mainly based on Perl and it is built around modules, which can connect to configuration files with webmin server. Thus, this gives the great advantage of adding new functionalities to the systems. By using Webmin, we can control many machines on the LAN with a single interface. It can also be configured to run under SSL. Basically, Webmin interface is intended to be used with root permissions but can be configured to be used by a group of users in the organization. But, to be more secure, the best practice is to give rights to only the root user for system administration. Thus, Webmin solves the problem of traditional console-based system administration and connection, through GUI in a web-based environment.

The main intention behind discussions about the Webmin tool in our thesis is to make it clear that enhancing security not only involves secure configurations but also to provide flexible and easy to use tools for system configuration.

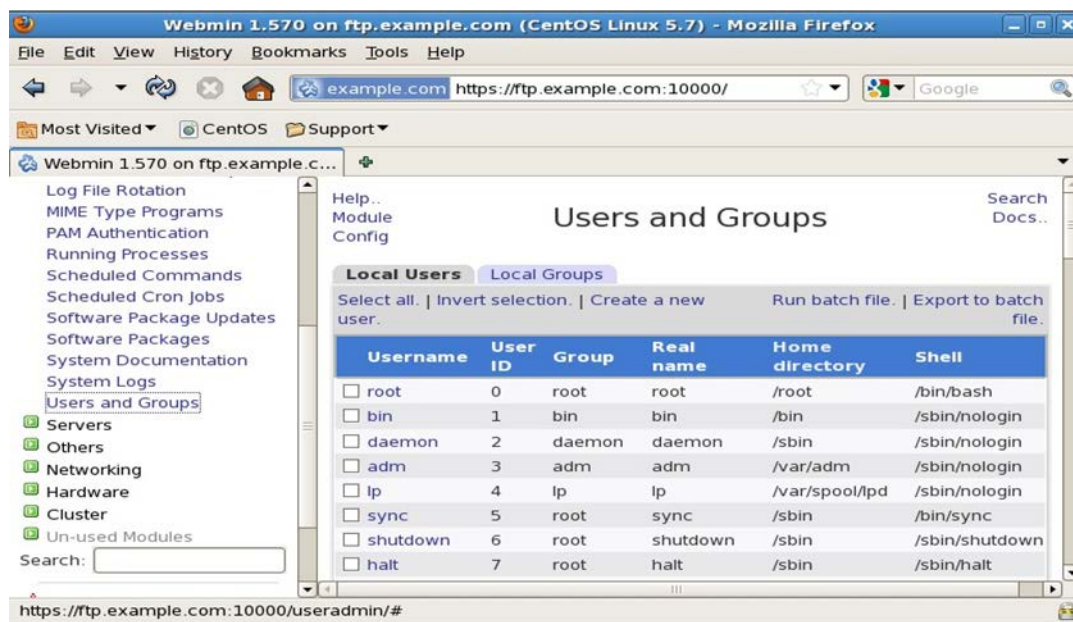


Figure 8. Webmin

## 2.8 Log Management in Linux

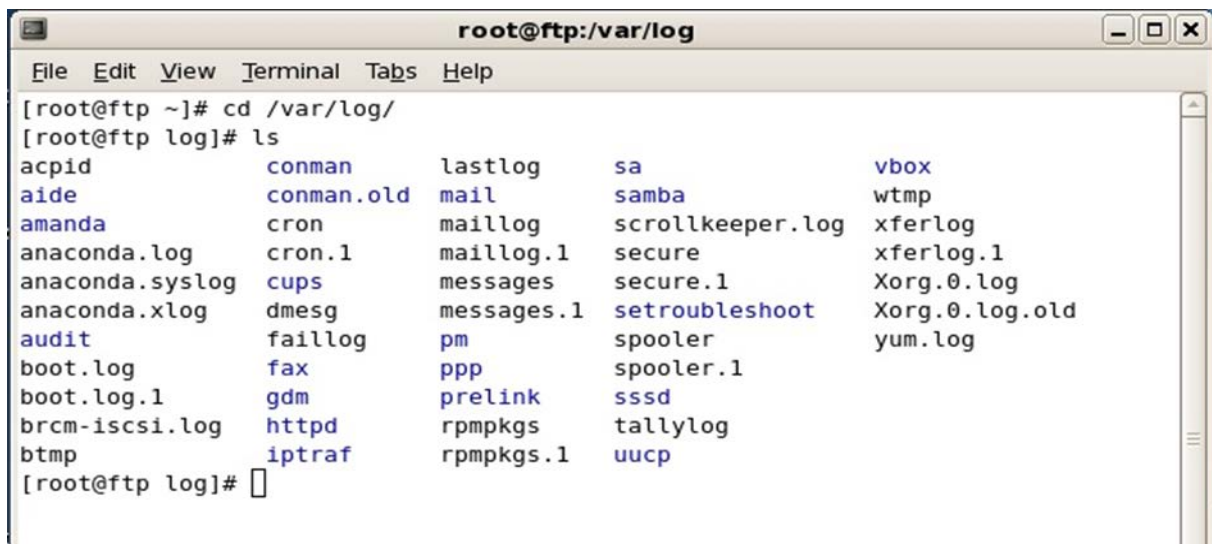
After understanding different network monitoring tools, the next important step is to get a clear understanding of inbuilt log management in Linux systems. Like network monitoring tools, Linux kernel possesses inbuilt system audit logs maintained in a separate folder /var/log, which records all the system activities. In the process of securing the system, one of the important steps is to have careful, accurate and comprehensive watched logs. The logs maintained by the system help the administrator in several ways, such as:

## Background

- Provide pre-damage warnings of a system.
- Provide valuable forensic data after a system crash or compromise.
- Helps to troubleshoot all types of system and application problems.

Log management or system auditing plays a key role in securing the services in the system. It will inspect record and audit all the security activities of the system [33]. Log check can also be configured to check regularly by using 'cron' utility. This helps the administrator to schedule the log management in his/her own convenient way. The main purpose of log check is to detect and prevent intrusions in the system by unauthorized users and to show the error operations of authorized users [34].

Linux possesses different utilities for log management such as tail, ps, netstat etc. Apart from all these utilities, Linux contains different log files in folder /var/log, which gives specific information of every single service accessed. Figure 9 demonstrates a sample output of contents in the folder /var/log.



```
root@ftp:/var/log
File Edit View Terminal Tabs Help
[root@ftp ~]# cd /var/log/
[root@ftp log]# ls
acpid          conman        lastlog       sa             vbox
aide           conman.old    mail          samba          wtmp
amanda         cron          maillog       scrollkeeper.log xferlog
anaconda.log   cron.1        maillog.1     secure         xferlog.1
anaconda.syslog cups          messages      secure.1       Xorg.0.log
anaconda.xlog  dmesg        messages.1    setroubleshoot Xorg.0.log.old
audit          faillog       pm            spooler        yum.log
boot.log       fax           ppp           spooler.1
boot.log.1     gdm          prelink       sssd
brcm-iscsi.log httpd         rpmpkgs       tallylog
btmp           iptraf       rpmpkgs.1    uucp
[root@ftp log]#
```

Figure 9. A sample showing output of folder /var/log

### 3 Experiment Setup

In this implementation part, one important thing to be considered is –'our implementation not only focuses on firewall configuration but also covers maximum aspects of building a secured network'. Therefore, the implementation part covers secured server installation and configurations, firewall configuration using IPtables, SE-Linux, etc. We have also applied some of the security measures in server configurations, which are discussed in the next section.

Figure 10 is the block diagram for implementation in a real environment, which clearly describes the function and position of each device in the network.

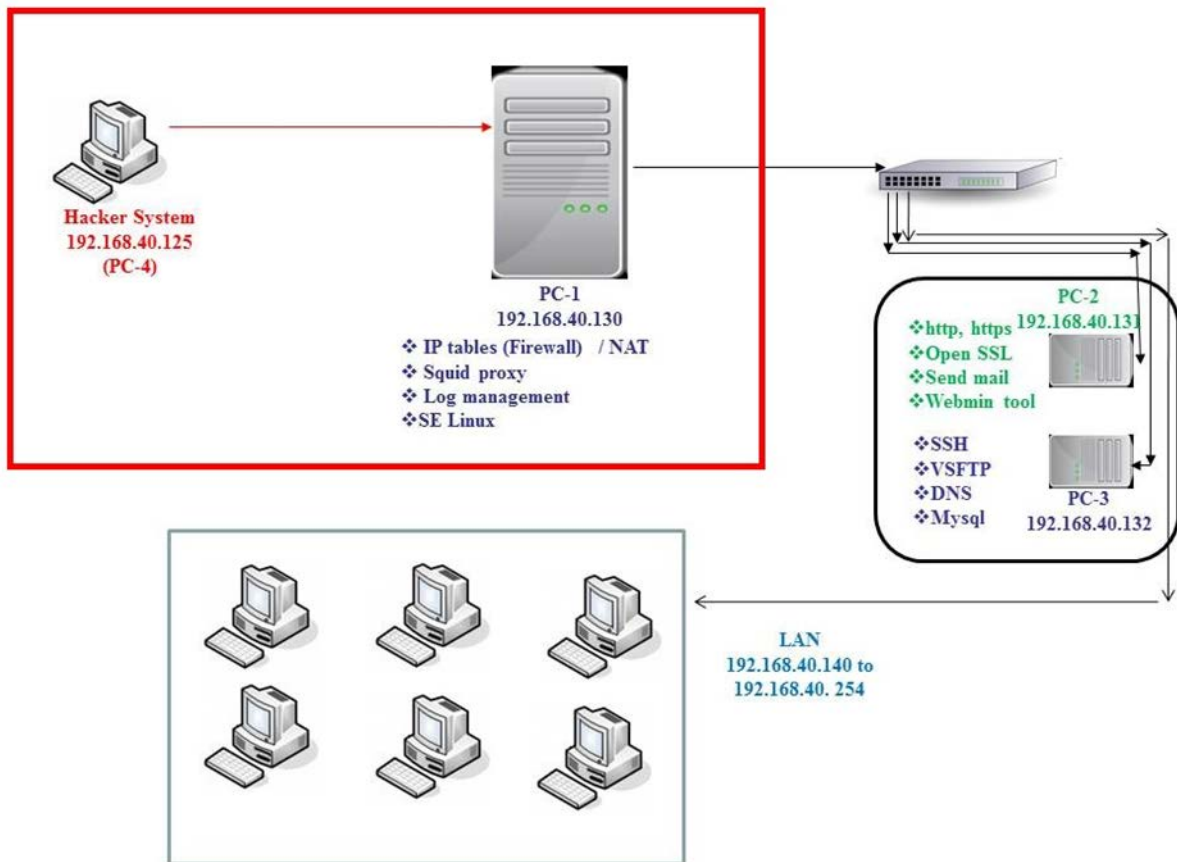


Figure 10. Block diagram for Implementation in real environment

The highlighted portion in the above diagram represents our implementation in VMware consisting of two systems, which are PC-1 and PC-4. All the configurations of PC-2 and PC-3 are added to PC-1 as we have considered only two systems in VMware instead of four systems.

**Note:** A detailed explanation of the procedure followed in obtaining the results is presented in chapter 1, methodology section of this report.



The configurations done in each system are discussed in the following section.

### **PC-1:**

#### **PC-1 consists of**

- Firewall configurations:
  - IPtables
  - SE Linux
- Network Monitoring tools:
  - Wireshark
  - IPtraf
- Server configurations
  - HTTP
  - HTTPS
  - OpenSSL
  - Sendmail
  - SSH
  - VSFTP
  - DNS
  - MySQL
- Web interface tool: Webmin

**PC-4:** PC-4 is used from outside network to test security of designed Linux network. This system contains few hacking tools to test the connectivity.

PC-4 consist of following tools

- Nmap
- Wireshark
- IPtraf

### **3.1 Security Measures Applied in Server Configurations**

Servers have a vital role in providing services to a computer in the network. In brief, they serve the needs of the user/client in the network. Depending on the computing services, the servers are classified as database servers, file servers, mail servers, web servers, etc. As these servers store the sensitive information, they need some security measures to be implemented.

#### **Apache web Servers:**

The main aim of an Apache web server is to establish secure communication between the server and client. Unsecured configurations of HTTP/HTTPS servers provide options for the hackers to gain access over the servers to create vulnerable services. To avoid these types of risks, we have presented some counter measures in Table 1.

SL.NO	Security Threats	Counter Measures
1	Attacker may generate vulnerable service in Document root file var/www/html	Change the document root directives file in httpd configuration
2	Hostname lookups in ON mode may create issues in the Internet	Do not set the hostname lookup option in ON mode in /etc/httpd/conf/httpd.conf
3	By default Document root is not configured to deny the request from any one.	Allow only trustworthy users in /etc/httpd/conf/httpd.conf file.
4	Enabling .htaccess file and .htpasswd in httpd configuration may lead to unsecured configuration.	Disable .htaccess file and .htpasswd in /etc/httpd/conf/httpd.conf file from being viewed by web clients.
5	Enabling memory map may reduce the performance of a server.	Disable the memory map option in OFF mode in /etc/httpd/conf/httpd.conf file

Table 1. Security Measures in Apache Web Servers

**FTP Server:**

The basic function of a FTP server is to transfer the files from servers to hosts in the network. If the FTP server is insecurely configured, it opens doors to the attacker, where the attacker pretends to be the authorized user to create vulnerable services. So, to avoid these types of security threats some counter measures are shown in Table 2, to secure the FTP server.

SL.NO	Security Threats	Counter Measures
1	By default anonymous user is enabled, which opens door for an attacker.	Restrict the anonymous users in /etc/vsftpd/vsftpd.conf file
2	FTP upload enable option can create vulnerable service.	Make anonymous upload disable in /etc/vsftpd/vsftpd.conf
3	Default greeting banner allows the attacker to determine usage of a system.	Change the default greeting banner in /etc/vsftpd/vsftpd.conf file.
4	Setting the ASCII mode in ON state may have the chance of malicious attacks.	Do not enable ASCII mode in /etc/vsftpd/vsftpd.conf file

Table 2. Security Measures in FTP Server

**DNS Server:**

The main function of a DNS Server is to translate the hostnames into IP addresses and vice versa. If the server is configured insecurely, then the attacker can constantly attempt to make the server unavailable which leads to the failure functioning of the server. So, to avoid these

types of security issues some of the counter measures are presented in Table 3 to secure the server.

SL.NO	Security Threats	Counter Measures
1	Enabling update option dynamically is a security risk factor in Zone Files.	Better to manually update the host information in /etc/named.rfc1912.zones
2	Bind weakness leads to the DNS vulnerability	Update the latest packages of Bind
3	Attacker can retrieve the list of IP address, if the zone transfer file is insecure.	Authenticate the zone transfer files in /etc/named.caching-nameserver.conf
4	DOS Attacks causes the web servers to be unavailable.	Install the network monitoring tools to monitor the attacks
5	Attacker can create, modify and delete the DNS record data if ACL is not configured in server.	Configure ACL to block any users to create, modify and delete DNS Data in /etc/named.caching-nameserver.conf
6	Disabling notify option helps the attacker to modify the data without the notice of Administrator.	Enable notify option in /etc/named.rfc.1912.zones to notify the messages whenever the zone changes.

**Table 3. Security Measures in DNS Server**

### SSH Server:

Generally, the SSH server's function is to provide secure communication over the Internet. Insecure server configuration leads to brute force attacks, which makes services vulnerable in the servers. So, to avoid these types of risks, we have presented some counter measures in Table 4 to secure the SSH server from the attackers.

SL.NO	Security Threats	Counter Measures
1	Insecure connection protocols makes SSH Server ineffective	Disable insecure connection protocols like Telnet and FTP
2	SSH protocol version 1 has security threats.	Use SSH protocol version 2 to avoid security risk factors in /etc/ssh/sshd_config
3	Allowing all users may have security threats in the server	Allow only trustworthy users in /etc/ssh/sshd_config
4	Disabling warning banner does not generate warning actions to users	Enable warning banner in /etc/ssh/sshd_config file

## Experiment Setup

5	Default ports are well known to the attackers	Change the default port numbers in /etc/ssh/sshd_config
6	By default users can access server directories such as /etc and /bin.	Protect the SSH server directories by using chroot or rssh tool.
7	Brute force attacks can be generated in SSH servers	To protect from Brute force attacks install Deny Host tool
8	Administrator does not have the knowledge of log information report, if the log is disabled.	Set the Log level option in Info mode in /etc/ssh/sshd_config
9	The out-dated patches leads to the security risk factors.	To enhance the security in SSH server and OS, update it with latest patches.

**Table 4. Security Measures in SSH Server**

### Sendmail Server:

The main function of Sendmail server is to accept and deliver the mails by using Mail User agent (MUA) and Mail Transfer Agent (MTA). Insecure configuration of Sendmail server leads to security risk factors such as retrieving the information of the mail, crashing the mail transfer agents, etc. So, to avoid these types of security threats we have applied some counter measures, which are shown in Table 5, to make the Sendmail server secure.

SL.NO	Security Threats	Counter Measures
1	Installing Sendmail with Set-user-id may exploit local vulnerabilities	Install Sendmail without set-user-id root
2	Attacker can crash the MTA by sending overloaded size file	Set the Maximum size limit value in /etc/mail/sendmail.mc
3	Attacker can retrieve the information easily if no encryption mechanism is configured	Configure SSL/TLS encryption mechanism in /etc/mail/sendmail.mc
4	Older version Sendmail has security bugs.	Update the latest versions of Sendmail server to avoid security bugs.
5	Advertising Sendmail version can create security issues	Do not advertise send mail version in /etc/mail/sendmail.mc

**Table 5. Security Measures in DNS Server**

## 4 Results

In order to have a clear understanding of the results obtained, we have illustrated the results in two scenarios. The results obtained **before** configuring firewall and applying security measures in server configurations are illustrated in **Scenario-1**, whereas the results obtained **after** configuring firewall and applying security measures in server configurations are explained in **Scenario-2**. The same steps are followed in both the scenarios in order to differentiate between the results obtained. Our main aim of showing the results in two scenarios is to have a clear distinction between the secured and unsecured systems. This distinction gives clear understanding of the security measures configured in the scenario-2.

Brief explanations of steps followed in obtaining results in both scenarios are:

**Step 1:** First, the target IP address is known by pinging the domain name of PC-1 from PC-4.

**Step 2:** Then, a reconnaissance attack is generated using nmap, to gather information of the target machine.

**Step 3:** Attack-related information is shown using IPtraf and Wireshark in PC-1.

**Step 4:** An attempt to access the HTTP server in PC-1 is done from PC-4.

**Step 5:** An attempt to access the FTP server in PC-1 is done from PC-4.

**Step 6:** The access information of different servers is shown as log messages in PC-1.

**Step 7:** The server access information is also shown in PC-1 using Wireshark.

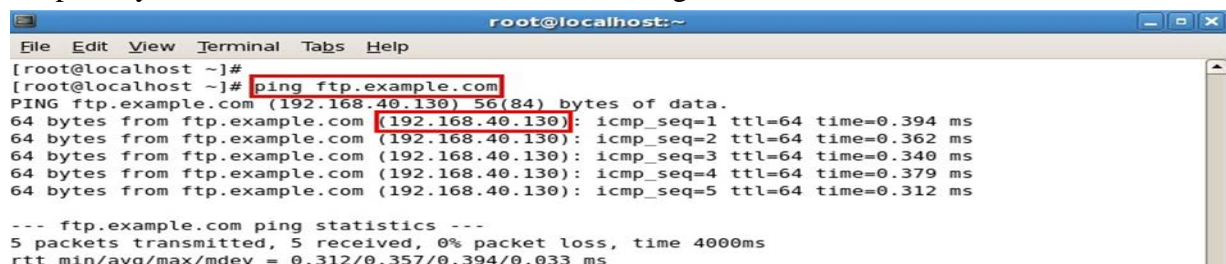
### 4.1 Results in Scenario-1

In Scenario-1, results obtained before configuring firewall and applying security measures in server configurations are explained.

Before configuring firewall and security measures on all the servers in PC-1, the system is insecure, allowing any user to access the information. The resources configured in PC-1 are easily accessed by the attacker system (PC-4). The results obtained in each of the above discussed steps are as follows.

*Step 1: First, the target IP address is known by pinging the domain name of PC-1 from PC-4.*

In order to attack a machine, the first thing is to know the IP address of the target machine. This can be accomplished by pinging the domain name of target machine, i.e., PC-1 from PC-4. When a ping command is executed in PC-4 by using the PC-1 domain name, the DNS server in PC-1 responds to the request made by the attacker system and sends the IP address of the machine as a reply message. There are also other ways to know the IP address using domain name, but using ping to know the IP address is the simplest and oldest method adopted by the attackers. This is illustrated in the Figure 11.



```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# ping ftp.example.com
PING ftp.example.com (192.168.40.130) 56(84) bytes of data.
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=1 ttl=64 time=0.394 ms
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=2 ttl=64 time=0.362 ms
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=3 ttl=64 time=0.340 ms
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=4 ttl=64 time=0.379 ms
64 bytes from ftp.example.com (192.168.40.130): icmp_seq=5 ttl=64 time=0.312 ms

--- ftp.example.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.312/0.357/0.394/0.033 ms

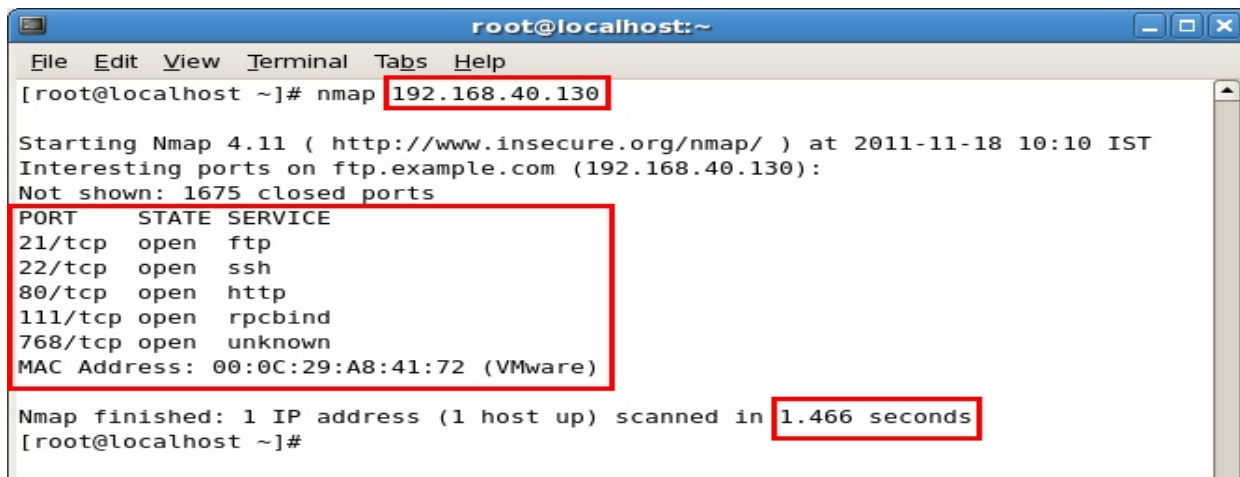
```

Figure 11. Finding the IP address using Ping

## Results

**Step 2:** Then, a reconnaissance attack is generated using *nmap*, to gather the information of the target machine.

After knowing the IP address of the target machine, then an attempt to gather the information of the target machine is performed using the *nmap* tool. This is known as a reconnaissance attack. This attack is used to know the services running on the target machines. This information is very useful for the attacker to know the vulnerabilities in the system. As PC-1 is an unsecured system, it is easy for the attacker to gather the information. The time taken to gather the information of an unsecured system will be reduced, which is shown in the Figure 12.



```
root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# nmap 192.168.40.130

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2011-11-18 10:10 IST
Interesting ports on ftp.example.com (192.168.40.130):
Not shown: 1675 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
768/tcp   open  unknown
MAC Address: 00:0C:29:A8:41:72 (VMware)

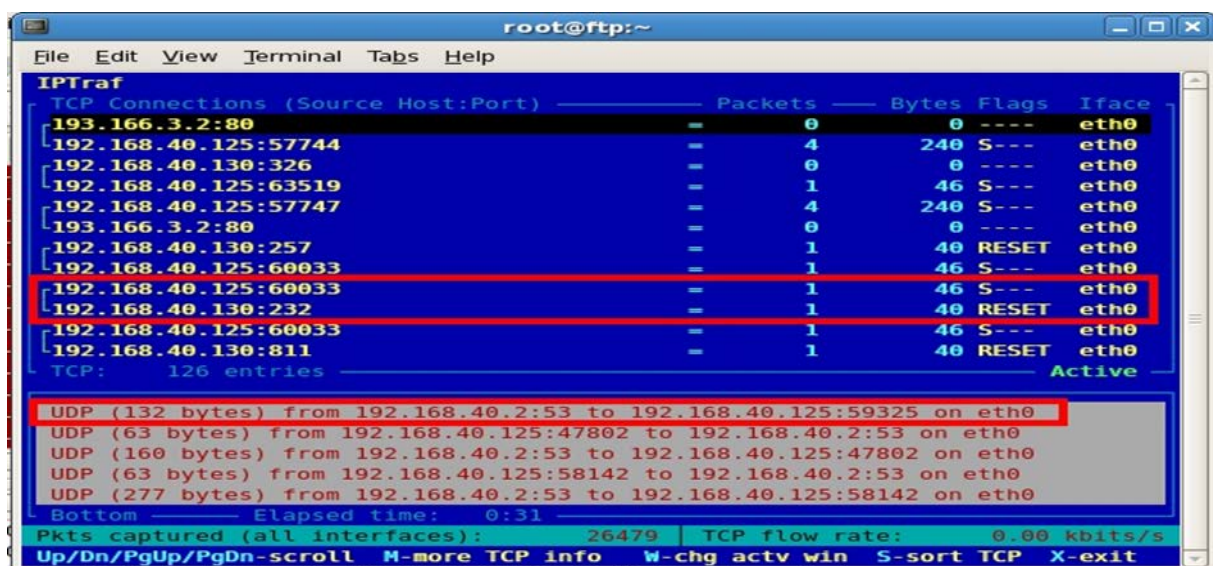
Nmap finished: 1 IP address (1 host up) scanned in 1.466 seconds
[root@localhost ~]#
```

Figure 12. The time taken by Nmap tool to access information of PC-1 before securing the system.

The time taken by the *nmap* tool to gather the information of the PC-1 before securing the system is only 1.466 seconds. This information gives the details of each service running on the PC-1. Hence, this information is very useful for the attacker to enter the system.

**Step 3:** Attack-related information is shown using *IPtraf* and *Wireshark* in PC-1.

Moreover, while an attacker is trying to access the information of the target system, we can see information about this attack in PC-1 using network monitoring tools such as *Wireshark* and *IPtraf*, which we used.



```
root@ftp:~
File Edit View Terminal Tabs Help
IPtraf
TCP Connections (Source Host:Port)  Packets  Bytes  Flags  Iface
193.166.3.2:80                      =        0      0  ----  eth0
192.168.40.125:57744                 =        4    240  S---  eth0
192.168.40.130:326                   =        0      0  ----  eth0
192.168.40.125:63519                 =        1     46  S---  eth0
192.168.40.125:57747                 =        4    240  S---  eth0
193.166.3.2:80                      =        0      0  ----  eth0
192.168.40.130:257                   =        1     40  RESET eth0
192.168.40.125:60033                 =        1     46  S---  eth0
192.168.40.125:60033                 =        1     46  S---  eth0
192.168.40.130:232                   =        1     40  RESET eth0
192.168.40.125:60033                 =        1     46  S---  eth0
192.168.40.130:811                   =        1     40  RESET eth0
TCP: 126 entries                      Active

UDP (132 bytes) from 192.168.40.2:53 to 192.168.40.125:59325 on eth0
UDP (63 bytes) from 192.168.40.125:47802 to 192.168.40.2:53 on eth0
UDP (160 bytes) from 192.168.40.2:53 to 192.168.40.125:47802 on eth0
UDP (63 bytes) from 192.168.40.125:58142 to 192.168.40.2:53 on eth0
UDP (277 bytes) from 192.168.40.2:53 to 192.168.40.125:58142 on eth0

Bottom ----- Elapsed time: 0:31 -----
Pkts captured (all interfaces): 26479 | TCP flow rate: 0.00 kbits/s
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit
```

Figure 13. Attack information in PC-1 using IPtraf in Scenario-1

## Results

In Figure 13, we can see that the attacker system from IP address 192.168.40.125 sends UDP packets to gather information of the target system 192.168.40.130 and we can see that the attacker uses different port numbers to connect the system. As PC-1 is unsecured, the attacker system has easily received a response to the requests made by the nmap tool. Figure 14 gives the same information using Wireshark.

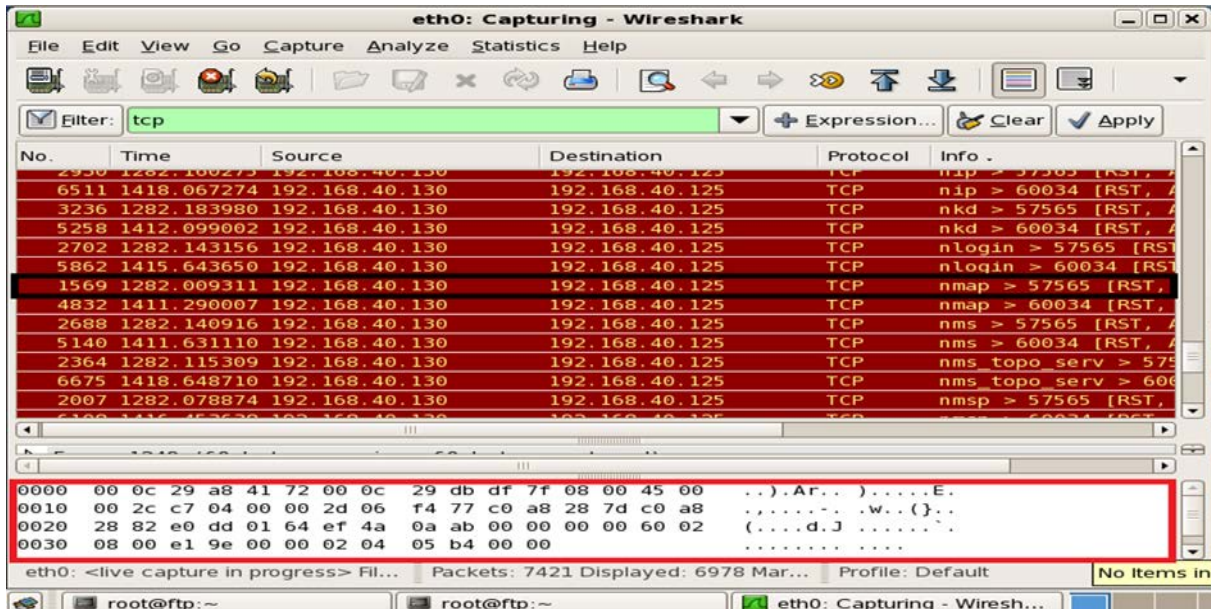


Figure 14. Attack information in PC-1 using wireshark in Scenario-1

From Figure 14, we can see that PC-1 (192.168.40.130) has successfully sent reply messages to the request made by the nmap tool in PC-4 (192.168.40.125).

**Step 4:** An attempt to access the HTTP server in PC-1 is done from PC-4.

After identifying the services offered in PC-1 using the nmap tool from PC-4, the next step is to check whether the services are accessed by the attacker. The services can be easily accessed in this scenario, as the system is insecure without any firewall configuration. In this step, an attempt to access the HTTP server is done from PC-4.

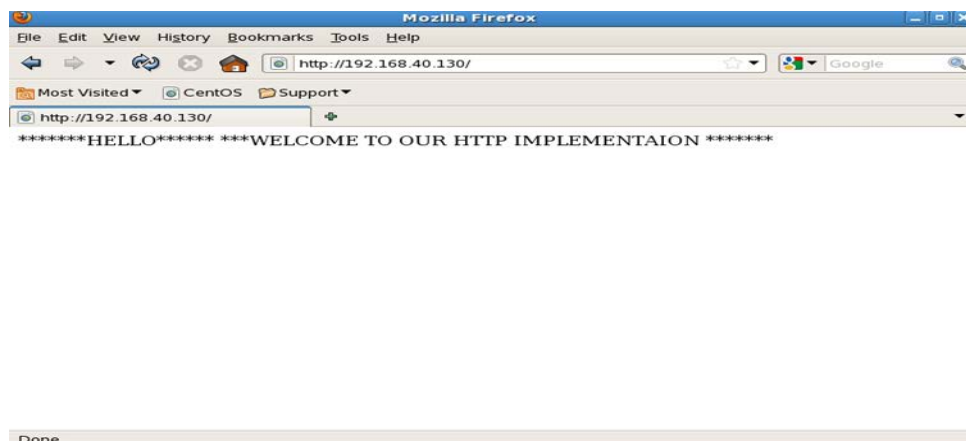


Figure 15. HTTP access from PC-4.

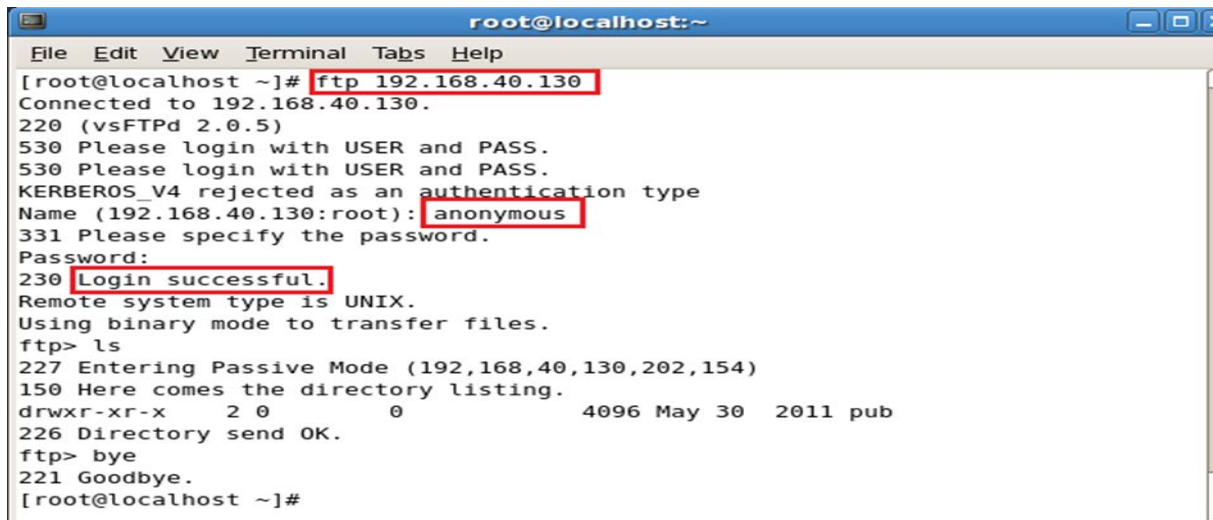
From Figure 15, we can see that the HTTP server is successfully accessed by the attacker system. Now, the attacker has gained access to the server and can perform malicious tasks to damage the server.



## Results

**Step 5:** An attempt to access the FTP server in PC-1 is done from PC-4.

After checking the HTTP access in step 4, an attempt to access the FTP server in PC-1 is performed to have a better understanding of the results in both the scenarios. The FTP server is also easily accessed by the attacker in the same way as the HTTP server in step 4 because of an unsecured system in this scenario. In Figure 16, the attacker has easily entered PC-1 using anonymous user in FTP. Once getting into the system, the attacker uses malicious programs to damage and gain access to the other hosts in the network. Therefore, this proves the level of insecurity in this scenario.

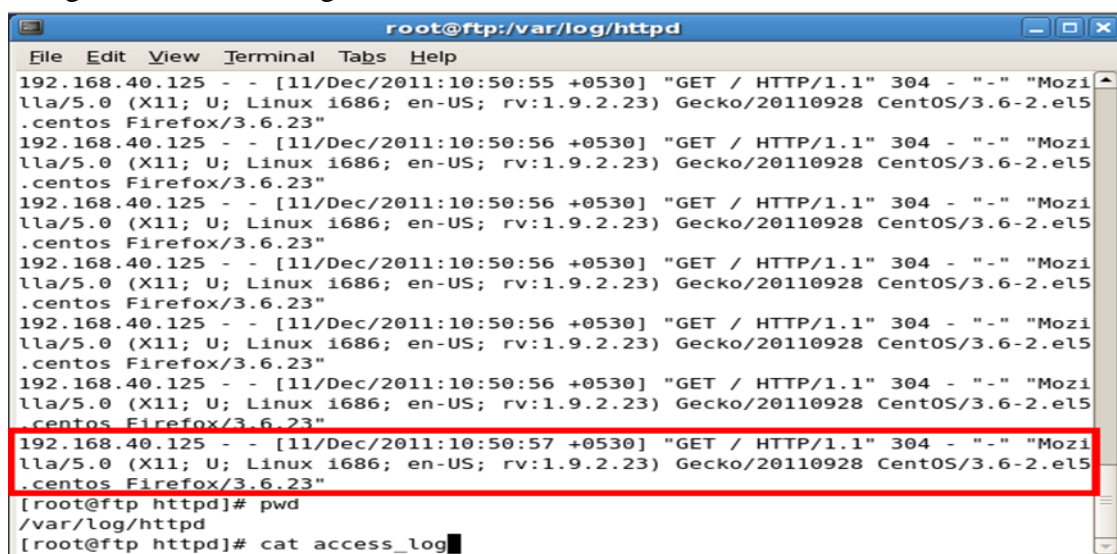


```
root@localhost:~# ftp 192.168.40.130
Connected to 192.168.40.130.
220 (vsFTPd 2.0.5)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (192.168.40.130:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,40,130,202,154)
150 Here comes the directory listing.
drwxr-xr-x  2 0          0               4096 May 30   2011 pub
ftp> bye
221 Goodbye.
[root@localhost ~]#
```

Figure 16. FTP access from PC-4.

**Step 6:** The access information of different servers is shown as log messages in PC-1.

The information related to the HTTP and FTP server access in step 4 and step 5 can be seen as log messages in /var/log folder. As the HTTP server is successfully accessed in step 4, this information is seen as log messages in /var/log/httpd/access\_log file. The information of access\_log file is shown in Figure 17.



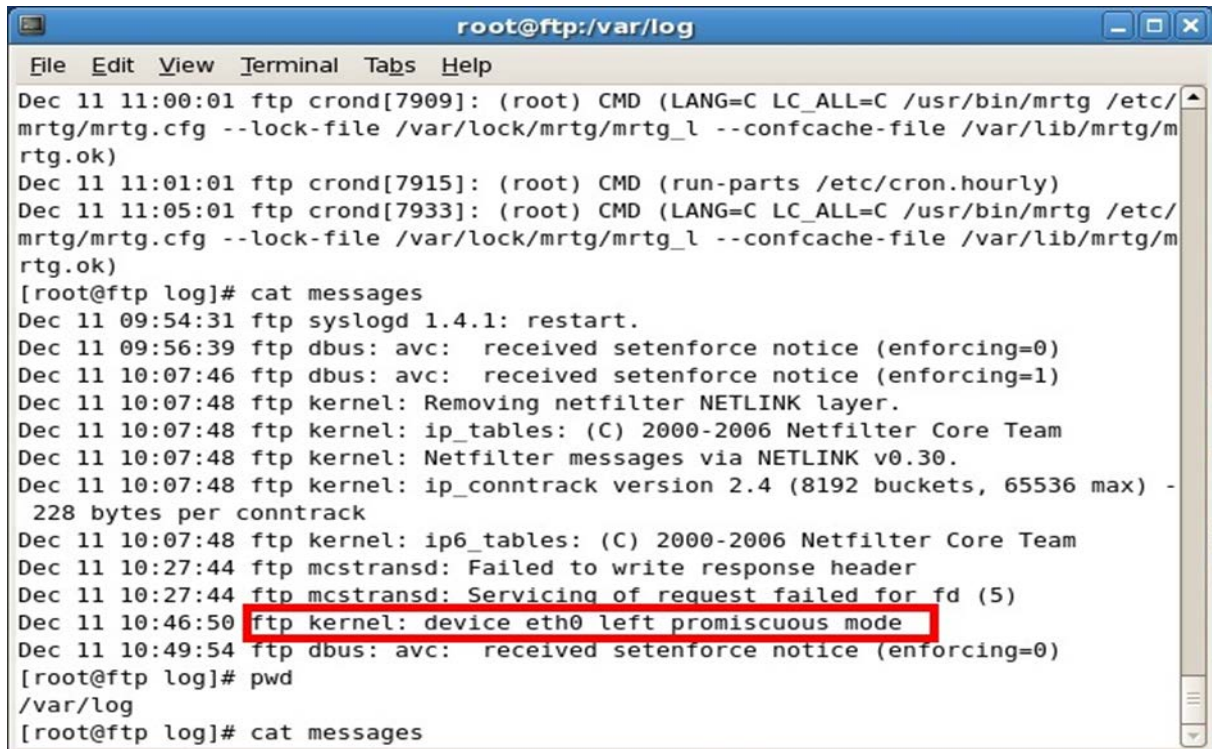
```
root@ftp:/var/log/httpd# cat access_log
192.168.40.125 - - [11/Dec/2011:10:50:55 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
192.168.40.125 - - [11/Dec/2011:10:50:56 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
192.168.40.125 - - [11/Dec/2011:10:50:56 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
192.168.40.125 - - [11/Dec/2011:10:50:56 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
192.168.40.125 - - [11/Dec/2011:10:50:56 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
192.168.40.125 - - [11/Dec/2011:10:50:57 +0530] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23) Gecko/20110928 CentOS/3.6-2.el5 .centos Firefox/3.6.23"
[root@ftp httpd]# pwd
/var/log/httpd
[root@ftp httpd]# cat access_log
```

Figure 17. Output showing the information of access\_log file in PC-1



## Results

In Figure 17, we can see that the HTTP server is successfully accessed from PC-4 (192.168.40.125). The detail information regarding HTTP access is found in this file. In step 5, the FTP server is accessed by the attacker system. This information is also seen as log message in /var/log/messages file, which is shown in the Figure 18.



```
root@ftp:/var/log
File Edit View Terminal Tabs Help
Dec 11 11:00:01 ftp crond[7909]: (root) CMD (LANG=C LC_ALL=C /usr/bin/mrtg /etc/
mrtg/mrtg.cfg --lock-file /var/lock/mrtg/mrtg_l --confcache-file /var/lib/mrtg/m
rtg.ok)
Dec 11 11:01:01 ftp crond[7915]: (root) CMD (run-parts /etc/cron.hourly)
Dec 11 11:05:01 ftp crond[7933]: (root) CMD (LANG=C LC_ALL=C /usr/bin/mrtg /etc/
mrtg/mrtg.cfg --lock-file /var/lock/mrtg/mrtg_l --confcache-file /var/lib/mrtg/m
rtg.ok)
[root@ftp log]# cat messages
Dec 11 09:54:31 ftp syslogd 1.4.1: restart.
Dec 11 09:56:39 ftp dbus: avc:  received setenforce notice (enforcing=0)
Dec 11 10:07:46 ftp dbus: avc:  received setenforce notice (enforcing=1)
Dec 11 10:07:48 ftp kernel: Removing netfilter NETLINK layer.
Dec 11 10:07:48 ftp kernel: ip_tables: (C) 2000-2006 Netfilter Core Team
Dec 11 10:07:48 ftp kernel: Netfilter messages via NETLINK v0.30.
Dec 11 10:07:48 ftp kernel: ip_conntrack version 2.4 (8192 buckets, 65536 max) -
228 bytes per conntrack
Dec 11 10:07:48 ftp kernel: ip6_tables: (C) 2000-2006 Netfilter Core Team
Dec 11 10:27:44 ftp mcstransd: Failed to write response header
Dec 11 10:27:44 ftp mcstransd: Servicing of request failed for fd (5)
Dec 11 10:46:50 ftp kernel: device eth0 left promiscuous mode
Dec 11 10:49:54 ftp dbus: avc:  received setenforce notice (enforcing=0)
[root@ftp log]# pwd
/var/log
[root@ftp log]# cat messages
```

**Figure 18. Output showing the information of messages file in PC-1**

From Figure 18, it is clear that the FTP server is accessed using eth0 interface. This interface eth0 is left in promiscuous mode while accessing the FTP server. The same information related to the access of different servers can also be seen using different network monitoring tools.

**Step 7:** *The server access information is also shown in PC-1 using Wireshark.*

In step 6, the information related to server access is shown using log messages, whereas in this step, the same information of server access is shown using wireshark. Our aim is to present our results in all aspects and to have a clear and detailed understanding of results. Figure 19 shows the server access information using wireshark in PC-1.

## Results

No.	Time	Source	Destination	Protocol	Info
32	27.589453	192.168.40.125	192.168.40.130	FTP	Request: SYST
33	27.589757	192.168.40.130	192.168.40.125	FTP	Response: 215 UNIX
34	27.629229	192.168.40.125	192.168.40.130	TCP	36828 > ftp [ACK]
35	30.054342	192.168.40.125	192.168.40.2	DNS	Standard query A f
36	30.078859	192.168.40.2	192.168.40.125	DNS	Standard query res
37	30.079351	192.168.40.125	130.226.184.9	TCP	51029 > ftp [SYN]
38	33.079246	192.168.40.125	130.226.184.9	TCP	51029 > ftp [SYN]
39	34.900702	192.168.40.125	192.168.40.130	TCP	34934 > http [SYN]
40	34.900968	192.168.40.130	192.168.40.125	TCP	http > 34934 [SYN]
41	34.901261	192.168.40.125	192.168.40.130	TCP	34934 > http [ACK]
42	34.903277	192.168.40.125	192.168.40.130	HTTP	GET / HTTP/1.1
43	34.903374	192.168.40.130	192.168.40.125	TCP	http > 34934 [ACK]
44	34.906820	192.168.40.130	192.168.40.125	HTTP	HTTP/1.1 304 Not M

Figure 19. Output showing the information of server access in PC-1 using wireshark

In Figure 19, we can see that PC-1(192.168.40.130) is exchanging packets with the attacker system (192.168.40.125) using HTTP and FTP servers.

The results discussed in all the seven steps in this scenario are explained using the secured system configuration in the next scenario.

## 4.2 Results in Scenario-2

In this Scenario, the results obtained after applying security measures in firewall and server configurations are explained.

PC-1 system is secured by configuring firewall and applying security measures to the server configurations. Now, the system restricts all outside users from accessing the resources. The results obtained in each of the seven steps are explained as follows.

**Step 1:** First, the target IP address is known by pinging the domain name of PC-1 from PC-4.

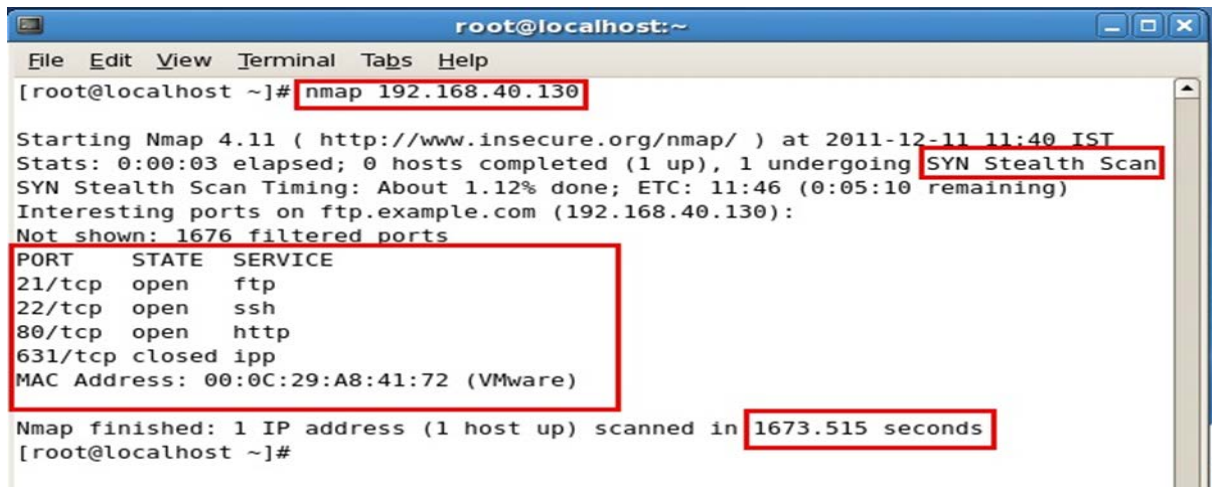
The results obtained in this step are same as the results in step-1 of scenario-1. The same method of ping is used by the attacker to know the IP address of PC-1 in this scenario. Hence, the explanation is identical to the explanation in step-1 of scenario-1.

**Step 2:** Then, a reconnaissance attack is generated using nmap, to gather the information of the target machine.

In this step, the same type of attack as in step 2 of scenario-1 is used to gather information. However, the results vary, because the system is secured now. It's very hard for the attacker to gather the information of a secured system. Sometimes, it is not even possible for the attacker to gather the information of a secured system. If possible, it takes a longtime to complete. The main difference lies in the time taken by the attacker system to gather the

## Results

information of the target machine. The time taken to gather the information of a secured system is shown in Figure 20.



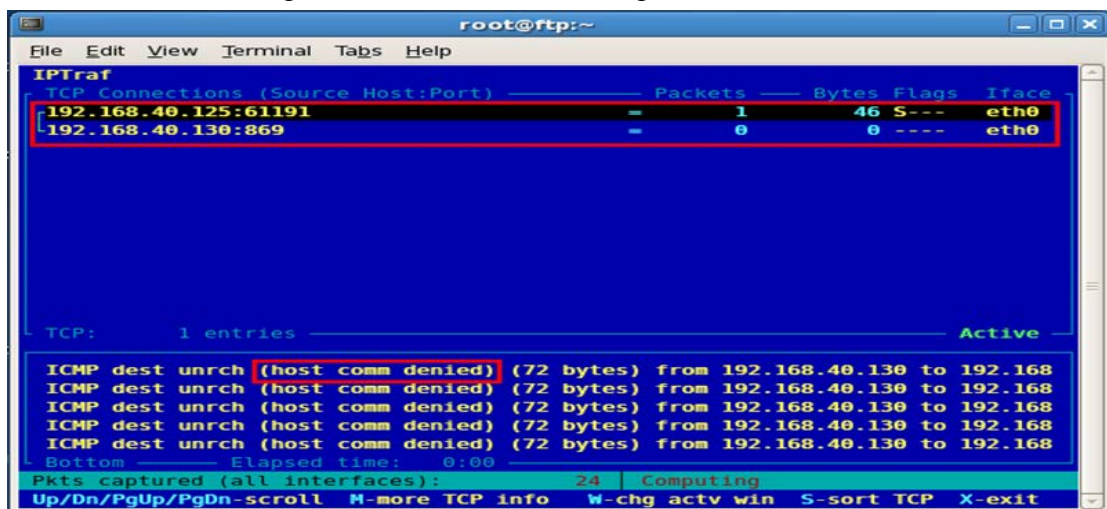
```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# nmap 192.168.40.130  
  
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2011-12-11 11:40 IST  
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan  
SYN Stealth Scan Timing: About 1.12% done; ETC: 11:46 (0:05:10 remaining)  
Interesting ports on ftp.example.com (192.168.40.130):  
Not shown: 1676 filtered ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
631/tcp   closed ipp  
MAC Address: 00:0C:29:A8:41:72 (VMware)  
  
Nmap finished: 1 IP address (1 host up) scanned in 1673.515 seconds  
[root@localhost ~]#
```

Figure 20. The time taken by nmap tool to access information of PC-1 after securing the system.

The time taken by the nmap tool after secure firewall configuration is 1673.515 seconds. This is because the scanning is held in stealth mode, as the firewall in PC-1 blocks the unauthorized access of the system information. This time increases as the number of hops between the target and attacker system increases. It may take days, weeks, months or years to gather the information depending on how far the target is from attacker and type of security measures applied on the target system.

**Step 3:** Attack-related information is shown using IPtraf and Wireshark in PC-1.

After generating the attack in step 2 using nmap tool, we can see the attack related information using any network monitoring tool in PC-1. As the system is secured now, PC-1 denies the request made by the attacker system to gather the information. This request-deny information is shown using IPtraf and wireshark in Figure 21.



```
root@ftp:~  
File Edit View Terminal Tabs Help  
IPtraf  
TCP Connections (Source Host:Port) Packets Bytes Flags Iface  
192.168.40.125:61191 1 46 S--- eth0  
192.168.40.130:869 0 0 ---- eth0  
  
TCP: 1 entries Active  
  
ICMP dest unrch (host comm denied) (72 bytes) from 192.168.40.130 to 192.168  
ICMP dest unrch (host comm denied) (72 bytes) from 192.168.40.130 to 192.168  
ICMP dest unrch (host comm denied) (72 bytes) from 192.168.40.130 to 192.168  
ICMP dest unrch (host comm denied) (72 bytes) from 192.168.40.130 to 192.168  
ICMP dest unrch (host comm denied) (72 bytes) from 192.168.40.130 to 192.168  
Bottom Elapsed time: 0:00  
Pkts captured (all interfaces): 24 Computing  
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit
```

Figure 21. Attack information in PC-1 using IPtraf in Scenario-2.

We can see that the attacker system from IP address 192.168.40.125 sends ICMP packets to gather the information of the target system 192.168.40.130 and also you can see that 'host access denied' message in PC-1 for the response to attack. In the same way as in scenario-1



## Results

the attacker uses different port numbers to gather the information. This information of attack can also be seen using wireshark, which is shown in Figure 22.

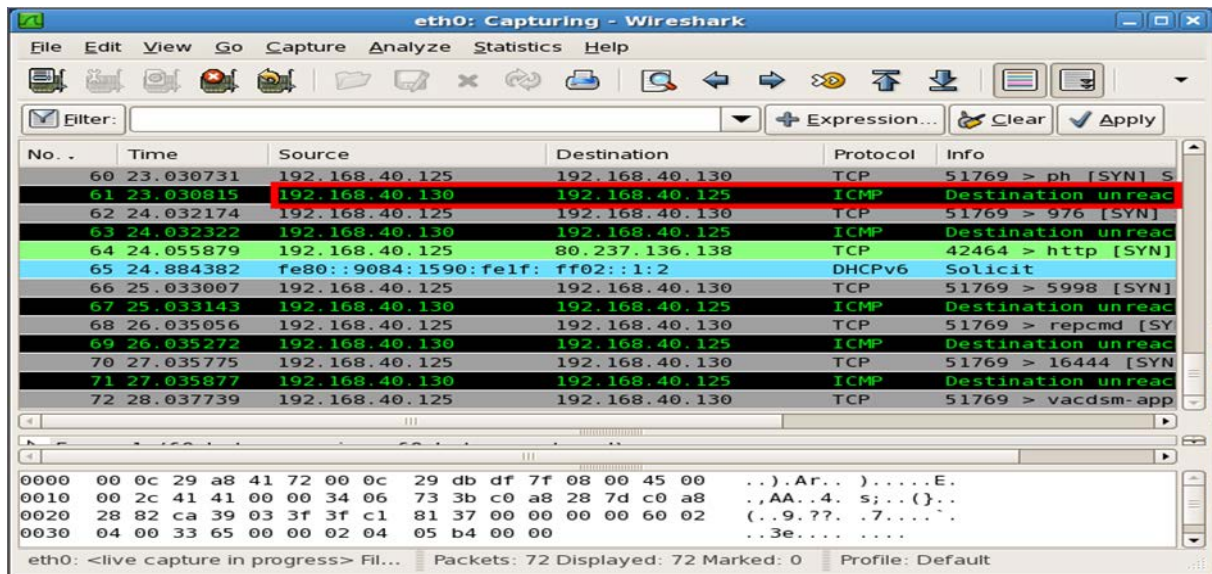


Figure 22. Attack information in PC-1 using Wireshark in Scenario-2.

From Figure 22, we can see that PC-1 (192.168.40.130) sends a reply as 'destination unreachable' to the request made by the attacker system (192.168.40.125).

**Step 4:** An attempt to access the HTTP server in PC-1 is done from PC-4.

After knowing the services offered in PC-1 using nmap tool from PC-4, the next step is to check whether the services are accessed by the attacker after configuring firewall and security measures in PC-1. The attacker cannot access the services in this scenario as the system is secured using a firewall. Figure 23 shows an attempt made by the attacker to access the HTTP server in PC-1.



Figure 23. HTTP access failure from PC-4.

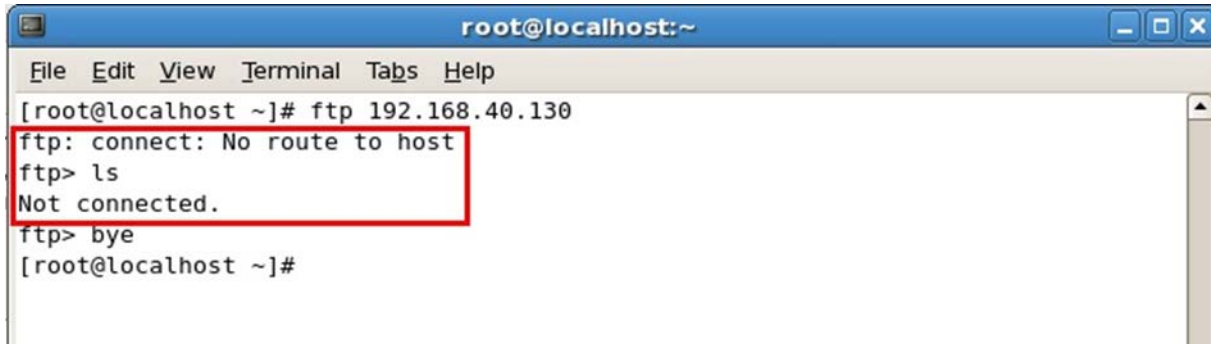
Figure 23 clearly explains that the attacker failed to access the HTTP server because the firewall in PC-1 prevents the unauthorized access of the servers, whereas it is easily accessed in step 4 of scenario-1, as the system in scenario-1 is not secured using any security measures. This proves the level of security applied to PC-1 in this scenario. Though the attacker

## Results

succeeded in gathering information of PC-1 but cannot access the servers running on PC-1. In the next step of this scenario, an attempt to access FTP is also shown.

**Step 5:** An attempt to access the FTP server in PC-1 is done from PC-4.

Even though, the security of PC-1 is proved in step 4 of this scenario, an attempt to access the FTP server in PC-1 is performed from PC-4 to have a clear understanding of the security configured in the system. As in the case of HTTP in step 4, the FTP access also failed due to the security configured in PC-1. This information of FTP access failure is shown in Figure 24.

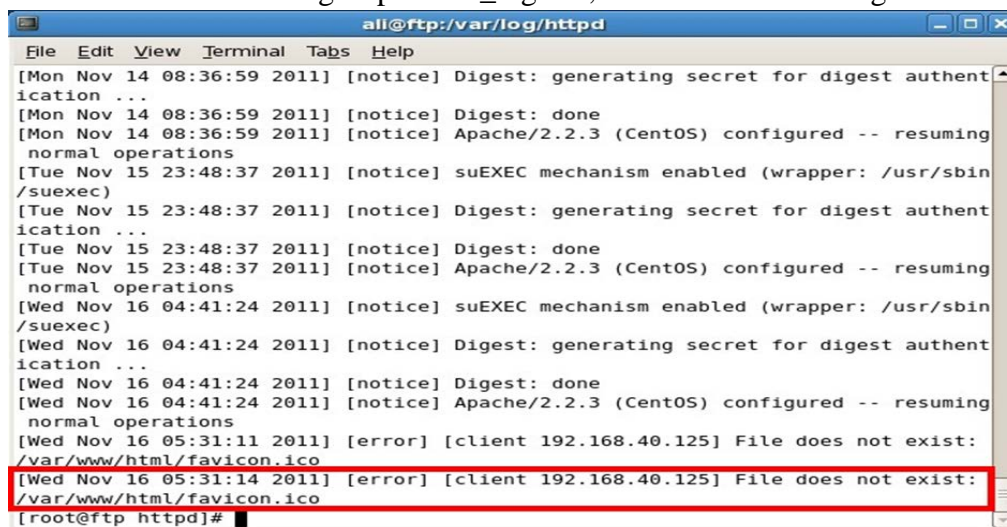
A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command '[root@localhost ~]# ftp 192.168.40.130' is entered. The output shows 'ftp: connect: No route to host', 'ftp> ls' returns 'Not connected.', and 'ftp> bye' returns '[root@localhost ~]#'. The first two lines are highlighted with a red box.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ftp 192.168.40.130  
ftp: connect: No route to host  
ftp> ls  
Not connected.  
ftp> bye  
[root@localhost ~]#
```

**Figure 24.** FTP access failure from PC-4.

**Step 6:** The access information of different servers is shown as log messages in PC-1.

In the above steps, it is clearly explained that the attempt to access the HTTP and FTP servers was not successful from the attacker system. This is because the firewall in PC-1 prevents unauthorized access of the servers. This information of trails done by the attacker system to access the HTTP and FTP services can be seen as the log messages in the PC-1 system. This step explains the server access failure information in log messages. The information of HTTP access failure is found in /var/log/httpd/error\_log file, which is shown in Figure 25.

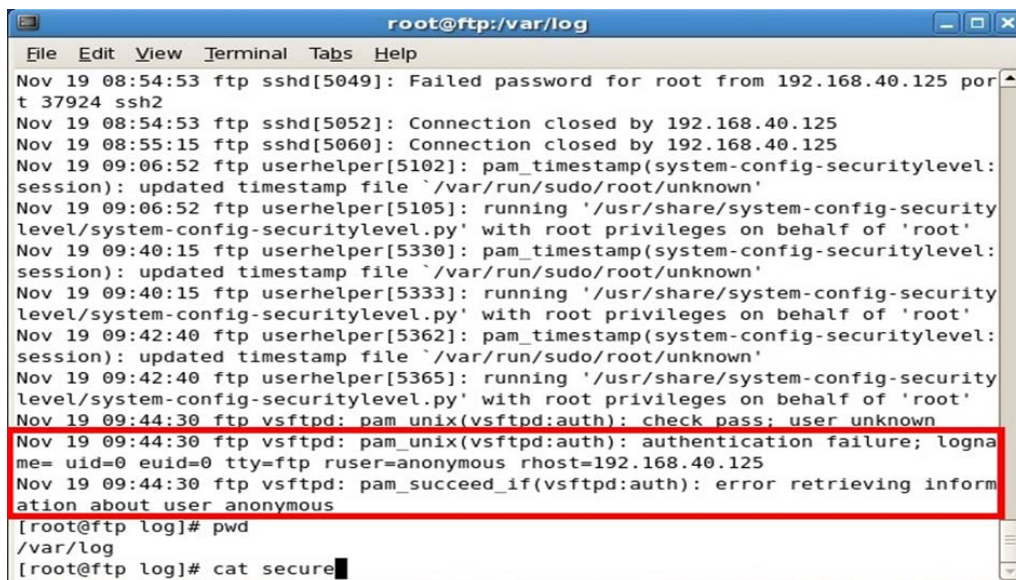
A terminal window titled 'ali@ftp:/var/log/httpd' with a menu bar (File, Edit, View, Terminal, Tabs, Help). It displays the contents of the /var/log/httpd/error\_log file. The log shows several 'notice' messages about Digest authentication and suEXEC mechanism. Two 'error' messages are highlighted with a red box: '[Wed Nov 16 05:31:11 2011] [error] [client 192.168.40.125] File does not exist: /var/www/html/favicon.ico' and '[Wed Nov 16 05:31:14 2011] [error] [client 192.168.40.125] File does not exist: /var/www/html/favicon.ico'. The prompt '[root@ftp httpd]#' is at the bottom.

```
ali@ftp:/var/log/httpd  
File Edit View Terminal Tabs Help  
[Mon Nov 14 08:36:59 2011] [notice] Digest: generating secret for digest authentication ...  
[Mon Nov 14 08:36:59 2011] [notice] Digest: done  
[Mon Nov 14 08:36:59 2011] [notice] Apache/2.2.3 (CentOS) configured -- resuming normal operations  
[Tue Nov 15 23:48:37 2011] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)  
[Tue Nov 15 23:48:37 2011] [notice] Digest: generating secret for digest authentication ...  
[Tue Nov 15 23:48:37 2011] [notice] Digest: done  
[Tue Nov 15 23:48:37 2011] [notice] Apache/2.2.3 (CentOS) configured -- resuming normal operations  
[Wed Nov 16 04:41:24 2011] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)  
[Wed Nov 16 04:41:24 2011] [notice] Digest: generating secret for digest authentication ...  
[Wed Nov 16 04:41:24 2011] [notice] Digest: done  
[Wed Nov 16 04:41:24 2011] [notice] Apache/2.2.3 (CentOS) configured -- resuming normal operations  
[Wed Nov 16 05:31:11 2011] [error] [client 192.168.40.125] File does not exist: /var/www/html/favicon.ico  
[Wed Nov 16 05:31:14 2011] [error] [client 192.168.40.125] File does not exist: /var/www/html/favicon.ico  
[root@ftp httpd]#
```

**Figure 25.** Output showing information of error\_log file in /var/log/httpd in PC-1

In Figure 25, the information related to HTTP access failure from the attacker system (192.168.40.125) is shown in error\_log file of PC-1. This file stores the information related to HTTP access failure. Alternatively, FTP access failure information is shown using /var/log/secure file in PC-1. This is shown in Figure 26.

## Results



```

root@ftp:/var/log
File Edit View Terminal Tabs Help
Nov 19 08:54:53 ftp sshd[5049]: Failed password for root from 192.168.40.125 port
t 37924 ssh2
Nov 19 08:54:53 ftp sshd[5052]: Connection closed by 192.168.40.125
Nov 19 08:55:15 ftp sshd[5060]: Connection closed by 192.168.40.125
Nov 19 09:06:52 ftp userhelper[5102]: pam_timestamp(system-config-securitylevel:
session): updated timestamp file '/var/run/sudo/root/unknown'
Nov 19 09:06:52 ftp userhelper[5105]: running '/usr/share/system-config-security
level/system-config-securitylevel.py' with root privileges on behalf of 'root'
Nov 19 09:40:15 ftp userhelper[5330]: pam_timestamp(system-config-securitylevel:
session): updated timestamp file '/var/run/sudo/root/unknown'
Nov 19 09:40:15 ftp userhelper[5333]: running '/usr/share/system-config-security
level/system-config-securitylevel.py' with root privileges on behalf of 'root'
Nov 19 09:42:40 ftp userhelper[5362]: pam_timestamp(system-config-securitylevel:
session): updated timestamp file '/var/run/sudo/root/unknown'
Nov 19 09:42:40 ftp userhelper[5365]: running '/usr/share/system-config-security
level/system-config-securitylevel.py' with root privileges on behalf of 'root'
Nov 19 09:44:30 ftp vsftpd: pam_unix(vsftpd:auth): check pass; user unknown
Nov 19 09:44:30 ftp vsftpd: pam_unix(vsftpd:auth): authentication failure; logna
me= uid=0 euid=0 tty=ftp ruser=anonymous rhost=192.168.40.125
Nov 19 09:44:30 ftp vsftpd: pam_succeed_if(vsftpd:auth): error retrieving inform
ation about user anonymous
[root@ftp log]# pwd
/var/log
[root@ftp log]# cat secure

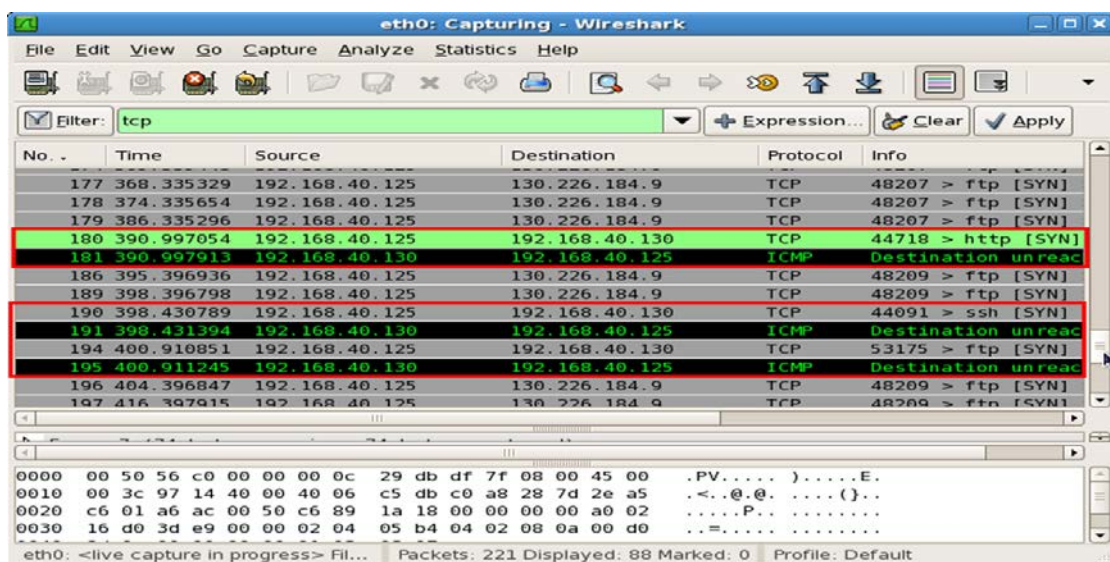
```

**Figure 26. Output showing the information of secure file in /var/log/ in PC-1**

In Figure 26, the highlighted information explains about the FTP access failure. The attacker system is trying to login FTP server using an anonymous user, but the system denies the request and prevents the FTP access by the attacker system (192.168.40.125). The level of security configured in PC-1 is proved again in this step. The information related to server access failure can also be seen using any network monitoring tools.

**Step 7: The server access information is also shown in PC-1 using Wireshark.**

In this step, the information related to server access failure is shown using wireshark in PC-1. This is shown in Figure 27.



No.	Time	Source	Destination	Protocol	Info
177	368.335329	192.168.40.125	130.226.184.9	TCP	48207 > ftp [SYN]
178	374.335654	192.168.40.125	130.226.184.9	TCP	48207 > ftp [SYN]
179	386.335296	192.168.40.125	130.226.184.9	TCP	48207 > ftp [SYN]
180	390.997054	192.168.40.125	192.168.40.130	TCP	44718 > http [SYN]
181	390.997913	192.168.40.130	192.168.40.125	ICMP	Destination unreach
186	395.396936	192.168.40.125	130.226.184.9	TCP	48209 > ftp [SYN]
189	398.396798	192.168.40.125	130.226.184.9	TCP	48209 > ftp [SYN]
190	398.430789	192.168.40.125	192.168.40.130	TCP	44091 > ssh [SYN]
191	398.431394	192.168.40.130	192.168.40.125	ICMP	Destination unreach
194	400.910851	192.168.40.125	192.168.40.130	TCP	53175 > ftp [SYN]
195	400.911245	192.168.40.130	192.168.40.125	ICMP	Destination unreach
196	404.396847	192.168.40.125	130.226.184.9	TCP	48209 > ftp [SYN]
197	416.307915	192.168.40.125	130.226.184.9	TCP	48209 > ftp [SYN]

**Figure 27. Output showing information of server access failure in PC-1 using wireshark**

In Figure 27, the highlighted information shows that the attacker tries to access the HTTP and FTP servers in PC-1 by sending packets. But we can see the 'destination unreachable' messages in response to the requests made by the attacker. The same step explained in scenario-1 gives the information related to the success of HTTP and FTP access from the

## Results

attacker system, whereas in this scenario, it is clear that the attacker failed in accessing HTTP and FTP server due to the security configured in PC-1.

### 4.3 Results Comparison

The results obtained in both the scenarios (i.e., Scenario-1 & 2) are compared in the following table.

Step	Scenario-1	Scenario-2
1	After pinging the domain name, IP address is known.	The information similar to Scenario-1 is known here.
2	Nmap gathers the information of the target machine in 1.466 seconds only. It is easy to gather the information.	Nmap gathers the information of the target machine in 1673.515 seconds which is taking large amount of time compared to scenario 1. It is very hard to gather the information, as the scanning is held in stealth mode.
3	Using IPtraf, we can see that the attacker system has easily received response to the requests made by the nmap tool and when working with Wireshark, it shows that attacker has successfully exchanged packets with the target machine.	In this scenario using IPtraf, we can see that the attacker has been successfully blocked from entering the target machine and using Wireshark, we can see that the attacker was not able to reach the target machine.
4	Attacker from PC-4 was able to access the HTTP server successfully.	Attacker was not able to access the HTTP server.
5	Attacker was able to access the FTP server as an anonymous user without any issues.	Attacker from PC-4 was not able to access the FTP server, i.e., the attacker was blocked from entering the system.
6	The information in /var/log/httpd/access_log & /var/log/messages tells that attacker has successfully accessed the HTTP and FTP servers.	The information in /var/log/httpd/error_log & /var/log/secure tells that attacker was blocked from entering the system.
7	The server access information is shown using wireshark. The information in wireshark tells that the HTTP and FTP servers are being accessed.	Based on information in Wireshark, the attacker was not able to access the HTTP and FTP servers.

**Table 6. Comparison of results in Scenario-1&2**

## **5 Conclusion**

From this implementation and research of enhancing network security, we found that; security is not only limited in choosing a secured operating system or secured server configurations, but also related to both physical and application security configured in the network. Moreover, periodical enhancement of network security is to be performed in order to get rid of day to day attacks. Servers which contain important information are to be configured securely and placed in a secured environment. The Firewall which works as the gateway for the network should be configured in such a way that it should not allow unauthorized users entering the network or accessing the information. Network audit information such as log messages and network monitoring tool's record will also help in securing the network by providing information about the network access. However, network security is a wide area of research in which policies and procedures used for security implementation are updated frequently, based on types of new attacks discovered.

Our implementation of enhancing network security can be extended by using any Intrusion Prevention System (IPS) or Intrusion Detection System (IDS). Use of IDS with any IPS is highly recommended, because IDS detects the attack whereas IPS can be used to prevent that particular attack.





## 6 References

1. Peter G.smith,"Linux Network Security", Charles River Media, Edition 1, March 2005.
2. Ken Denniston, "Building a Simple Network", Intel Press, Edition 2, Chapter -1.
3. Wenzheng Zhu; Changhoon Lee; Coll. of Comput., Konkuk Univ., Seoul", Design for Security Operating System", IEEE Computer Conference on Third Asia International, pp. 667-670, 2009.
4. LI Hongjuan, LAN Yuqing, "A Design of Trusted Operating System Based on Linux",IEEE Computer International conference on Electrical and control Engineering, pp 4598 – 4601,2010.
5. Bokhari, S.N.,"The Linux operating system", IEEE Computer,vol. 28,no. 8,pp 74-79.1995.
6. W. A. Arbaugh, D. J. Farber, and J. M. Smith, "A Secure and Reliable Bootstrap Architecture," in IEEE Computer Society Conference on Security and Privacy. IEEE, 1997, pp. 65–71.
7. Mark G. Sobell, "A Practical Guide to Ubuntu Linux", Third Edition, Pearson.
8. Haral Tsitsivas, "UNIX System Management and Security: Differences between Linux, Solaris, AIX and HP-UX", white paper, SANS institute, 2007.
9. Machtelt Garrel, "Introduction to Linux", Edition 1.27, 2008.
10. Jichiang Tsai ; Chung-Hsin Feng ; Chuyuan Tsai," A Network Safety-Defense Mechanism with the Linux Security Module",2006 IEEE Region 10 Conference ,pp. 1-4,2006.
11. Si-Jung Kim ; Choul-Woong Son ; Cheon-Woo Lee," Linux based Unauthorized Process Control"IEEE Computer Conference on ICISA,pp. 1-5,2011.
12. Chris Wright,Crispin Cowan, Stephen Smalley, James Morris, Greg Kroah-Hartman,"Linux Security Module:General Security Support for the Linux Kernel", Emmanuel Fleury, 2006-2007.
13. James Morris,"SELinux", source: [http://selinuxproject.org/page/Main\\_Page](http://selinuxproject.org/page/Main_Page)(Last accessed: February 06, 2012)
14. AlanBartlett, "SELinux", Source: <http://wiki.centos.org/HowTos/SELinux> (Last accessed February 06, 2012)
15. Werner Puschitz," Securing and Hardening Red Hat Linux Production Systems", PUSCHITZ.COM, 2007.
16. J. Marchesini, S. Smith, O. Wild, and R. MacDonald, "Experimenting with TCPA/TCG Hardware, Or: How I Learned to Stop Worrying and Love the Bear," in Technical Report TR2003-476, Dartmouth PKI Lab Dartmouth College, Hanover, New Hampshire, USA, December 2003.

17. Deng Yiquan, "Linux Network Security Technology", IEEE Computer Conference on CASE, pp. 1-3, 2011.
18. William Stallings, "Cryptography and Network Security Principles and Practices", Edition:4, Prentice Hall, 2005.
19. Eric A. Young, Tim J. Hudson, "Open SSL", Source: <http://www.openssl.org/> (Last accessed: February 06, 2012).
20. Anthony J. Stieber, "OpenSSL Hacks", Linux Journal Issue #147/July 2006.
21. Ubuntu documentation team, "Open SSH Server"  
Source: <https://help.ubuntu.com/8.04/serverguide/C/openssh-server.html> (Last accessed: February 06, 2012).
22. R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, "DNS Security Introduction and Requirements", rfc: 4033, The Internet Society, 2005.
23. Duane De Capite, "Self-Defending Networks: The Next Generation of Network Security", Cisco Systems, Inc., September 2006.
24. William H. Allen, Gerald A. Martin and Luis A. Rivera. "Automated detection of malicious reconnaissance to enhance network security" in IEEE conference on South east, Publication 2005, pages: 450-454.
25. David Kotfila, Joshua Moorhouse, Ross Wolfson, "CCNP Implementing Secured Converged Wide-Area Networks (ISCW 642-825) Lab Portfolio. (Last accessed: November 24, 2011 )
26. IBM Corporation "Understanding DNS Queries", iSeries Information Center, Version 5 Release 3, 2002, 2005.
27. Rick Hofstede, Tiago Fioreze, "Surf Map: A network Monitoring Tools Based on the Google Maps Api" in IEEE conference on Integrated Network Management, Publication 2009, Pages: 676-690.
28. Paul Ferrill "Linux Network Monitoring Tools -Ping and Etherape", Tutorial, QuinStreet Inc, 2012.
29. Gordon Lyon, "Nmap.org", Source: <http://nmap.org/> (Last accessed: February 06, 2012).
30. Richard Sharpe, Ed Warnicke "Wireshark",  
Source: [http://www.wireshark.org/docs/wsug\\_html/#ChapterIntroduction](http://www.wireshark.org/docs/wsug_html/#ChapterIntroduction) (Last accessed: February 06, 2012).
31. "IP Traf", Source: <http://iptraf.seul.org/index.html> (Last accessed: February 06, 2012).
32. "Webmin", Source: <http://www.webmin.com/intro.html> (Last accessed: February 06, 2012).
33. Michael D. Bauer, "Building Secure Servers with Linux", O'Reilly, ISBN:0-596-00217-3, 2002.
34. Kuo Zhao; Qiang Li; Jian Kang; Dapeng Jiang; Liang Hu, "Design and Implementation of Secure Auditing System in Linux Kernel", IEEE Computer, pp. 232-236, 2007.

## Appendix

### Configurations:

#### PC-1 :

##### Configuring: IP tables

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0

eth0 = 192.168.40.125

# cat /etc/sysconfig/network-scripts/ifcfg-eth1

eth0 = 192.168.40.130

# nano /etc/sysconfig/network (configuring gateway)

NETWORKING = yes
HOSTNAME= localhost
GATEWAY=192.168.40.30
# nano /etc/resolve.conf (providing name server IP address)
nameserver 192.168.40.130
# iptables -f (flushing all the rules in filter and nat tables)
# iptables -t nat -f
# iptables -F
# iptables -X
# iptables -t nat -X (deleting all chains that are not filtered at table)
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -t forward -i eth1 -j ACCEPT
# nano /etc/sysconfig/iptables (allows these port no for incoming)
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 10000 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT
# service iptables restart
# iptables -A INPUT -p tcp -s 0/0 --sport 1024:65535 -d $SERVER_IP --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT (accepting incoming http access)
# iptables -A OUTPUT -p tcp -s $SERVER_IP --sport 80 -d 0/0 --dport 1024:65535 -m state -state ESTABLISHED -j ACCEPT (accepting outgoing http access)
```

## Enhancing Network Security in Linux Environment

```
#iptables -A INPUT -p tcp -s 0/0 --sport 1024:65535 -d $SERVER_IP --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT (accepting incoming https access)
```

```
iptables -A OUTPUT -p tcp -s $SERVER_IP --sport 443 -d 0/0 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT (accepting outgoing https access)
```

```
# service iptables restart
```

Note: In the same way, all the service port numbers incoming and outgoing should be added in the Iptables, in order to filter the packets.

### Configuring: SE-Linux

SELinux main config file is /etc/selinux/config

```
# nano /etc/selinux/config
```

Change the status of

SELINUX=enable

```
# getenforce (to get the status)
```

```
# setenforce Permissive (making selinux permissive)
```

```
# getsebool -a
```

```
# setsebool - p allow_ftp_d_anon_write 1
```

```
# getsebool -a | grep allow_ftp_d_anon_write
```

```
# setsebool - p allow_ftp_d_anon_write 0
```

```
# getsebool -a | grep allow_ftp_d_anon_write
```

```
#sealert -b (log info)
```

```
#seinfo (status of selinux)
```

```
#system-config-selinux (graphical mode configuration of SELinux)
```

### Example of blocking

```
http_access deny google.com
```

```
http_access deny yahoo.com
```

```
http_access allow xyz
```

```
http_access allow google
```

**Note the above lines should be declared before this line**

```
http_access deny all
```

```
# service squid restart
```

```
# chkconfig squid on
```

Note: add squid proxy in iptables

```
# vi /etc/sysconfig/iptables
```

## Enhancing Network Security in Linux Environment

```
-A RH-Firewall-1-INPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp --dport 3128 -j ACCEPT
```

Restart iptables based firewall:

```
# /etc/init.d/iptables restart
```

```
# netstat -tulpn | grep 3128 ( verifying port 3128 is listening)
```

### Configuring http server:

To install : # rpm -ivh http\*

```
# nano /etc/httpd/conf/httpd.conf
```

```
ServerName ftp.example.com
```

```
DocumentRoot /var/www/html
```

```
DirectoryIndex file index.html
```

```
# httpd -t (to check the syntax)
```

```
# service httpd restart
```

```
# httpd -M or apache2ctl -M ( to check loaded modules)
```

### Configuring https server with SSL:

To install: # rpm -ivh mod\* ssl\*

```
# cd /etc/pki/tls/certs
```

```
# openssl genrsa -des3 -out apachekey.pem 2048 (generating apachekey.pem)
```

It will generate a key

```
# # openssl req -new -key apachekey.pem -out apachekey.csr (generating a signed certificate)
```

It will prompt many questions like name, web address and other information, and have to provide this information.

Creating web server certificate

```
# openssl ca -in apachekey.csr -out apachecert.pem
```

Installing or creating SSL certificate

```
# cp apachecert.pem /etc/pki/tls/http/
```

```
# cp apachekey.pem /etc/pki/tls/http/
```

```
# nano /etc/httpd/conf.d/sslconf (Open the SSL main config file)
```

```
Listen 192.168.0.100:443 (https server ip)
```

```
SSLRandomSeed startup file:/dev/urandom 1024
```

```
SSLRandomSeed connect file:/dev/urandom 1024
```

```
<VirtualHost ftp.exmple.com:443>
```

```
    SSLEngine On
```

```
    SSLCertificateFile /etc/pki/tls/http/apachecert.pem
```

## Enhancing Network Security in Linux Environment

```
SSLCertificateKeyFile /etc/pki/tls/http/apachekey.pem
SSLProtocol All -SSLv2
SSLCipherSuite HIGH:MEDIUM:!aNULL:+MD5
DocumentRoot "/var/www/html/ssl"
ServerName ftp.example.com:443
</VirtualHost>

# mkdir -p /var/www/html/ssl
# vi /etc/httpd/conf/httpd.conf
Add this at end of the following of file
<Directory /var/www/html/ssl>
    SSLRequireSSL
    SSLOptions +StrictRequire
    SSLRequire %{HTTP_HOST} eq "example.com"
    ErrorDocument 403 https://example.com/sslerror.html
</Directory>

# service httpd restart
```

### Configuring sendmail server

```
# rpm -ivh sendmail* m4* or # yum install sendmail* m4* -y
# nano /etc/mail/sendmail.mc (edit main config file)
Add
DNL # DAMON_OPTION ('port=smtp,adr=127.0.0.1, name=MTA') dnl
LOCAL_DOMAIN('SERVER.EXAMPLE.COM')dnl
# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf (complaining the file)
# service send mail restart
```

### Configuring SSH:

```
# nano /etc/ssh/sshd_config (main configuration file)

PermitRootLogin no (disabling root user login)
UsePrivilegeSeparation yes
Protocol 2 (prefer version 2only)
AllowTcpForwarding no (tcp port disabling in xinetd)
X11Forwarding no (tcp port disabling in xinetd)
StrictModes yes (generating key file and making secured)
IgnoreRhosts yes
```

## Enhancing Network Security in Linux Environment

HostbasedAuthentication no (stopping host based authentication)

RhostsRSAAuthentication no

#Subsystem sftp /usr/lib/misc/sftp-server (better stop always sftp for security reason)

# Service sshd restart (restart SSH service)

#nano /etc/sysconfig/iptables

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT

# service iptables restart.

### Configuring FTP Server:

# rpm -ivh vsft-2.0 ( to install vsftp)

# nano /etc/vsft/vsftpd.conf

anonymous\_enable=NO (disable anonymous user login to make it more secure)

chroot\_local\_user=YES (enable all the users and restricting their home directories)

userlist\_disable=NO

# service vsftpd restart

If we want restrict some users then

# nano /etc/vsftp/ftpusers

User1 (Add the user in this file)

# chmod 744 /var/ftp/pub (assigning few permission)

### Configuring DNS server:

Three packages are to be configured for DNS.

- bind
- bind-chroot
- caching-nameserver

Main configuration files are :

/etc/named.rfc1912.zones

/etc/named.caching-nameserver.conf

Installing:

#rpm -ivh bind-9.3.6-16.p1.e15.1386.rpm bind-chroot-9.3.6-16.p1.e15.1386.rpm caching-nameserver-9.3.6-16.p1.e15.1386.rpm

Configuring:

#hostname [ftp.example.com](http://ftp.example.com)



## Enhancing Network Security in Linux Environment

To make this host as permanent: #nano /etc/sysconfig/network

HOSTNAME=ftp.example.com (save and exit)

#nano /etc/resolv.conf (resolves your domain name into ip address)

search example.com

nameserver 192.168.40.130 (save and exit)

#nano /etc/hosts

Just add the following line:

192.168.40.130 [ftp.example.com](http://ftp.example.com) (save and exit)

#nano /etc/named.caching-nameserver.conf

Listen-on port 53 { 127.0.0.1; 192.168.40.130};

match-clients {localhost; 192.168.40.130};

match-destinations {localhost; 192.168.40.130};

(save and exit)

#nano /etc/named.rfc1912.zones

Zone “example.com” IN {

type master;

file “localhost.zone”

zone “40.168.192. in-addr.arpa” IN {

type master;

file “named.local”;

#nano localhost.zone

@ IN SOA [ftp.example.com](http://ftp.example.com)

IN NS [ftp.example.com](http://ftp.example.com) root.

ftp IN A 192.168.40.130

#nano named.local

@ IN SOA [ftp.example.com](http://ftp.example.com) root.example.com. (

## Enhancing Network Security in Linux Environment

```
IN      NS      ftp.example.com
130     IN      PTR      ftp.example.com
#named-checkzone      example.comlocalhost.zone
#named-checkconf      /etc/named.rfc1912.zones
#named-checkconf      /etc/named.cachingnameserver.conf
#service named restart
#service network restart
```

### **Configuring MySQL server:**

Install :

```
#yum install mysql* -y
```

```
#service mysqld start
```

Create a user in the database:

```
#mysqladmin -u root password '123'
```

### **PC-4:**

Install Nmap:

```
#rpm -ivh nmap-4.11-1.1.rpm
```

Install wireshark:

```
#rpm -ivh wireshar-1.0.15-1.e15_6.4
```

Install IPtraf:

```
#rpm -ivh iptraf-3.0.0-5.e15
```

### **List of Configured Files**

/etc/sysconfig/iptables  
/etc/vsftp/vsftpd.conf  
/etc/sysconfig/sshd\_conf  
/etc/httpd/conf/httpd.conf  
/etc/sysctl.conf  
/etc/init.d  
/etc/host.deny  
/etc/host.allow  
/etc/sysconfig/network-scrpits/ifcfg-eth  
/etc/reolve.conf  
/etc/hosts  
/etc/selinux/config  
/etc/squid/squid.conf  
/var/log/messages  
/etc/vsftp/ftpusers  
/etc/httpd/conf.d/sslconf  
/etc/mail/sendmail.mc  
/etc/mail/sendmail.cf  
/etc/password  
/etc/shadows  
/etc/ipforward