Program - 2

Create a LAN using physical networks/virtual machine and install FTP server to demonstrate file transfer

Description

Design a network topology with two nodes PC1 and PC2. The node PC1 and PC2 is connected to a 8 port switch in the center as shown in Figure 17. Assign IP address to each node and implement File Transfer Protocol. ***File Transfer Protocol***

FTP stands for **File Transfer Protocol**, and is a connection method designed for transferring files from a remote server to local computer, and vice versa. FTP is often used in corporate and academic settings, and is the primary way of managing webpage servers. File Transfer Protocol is represented in Figure 16

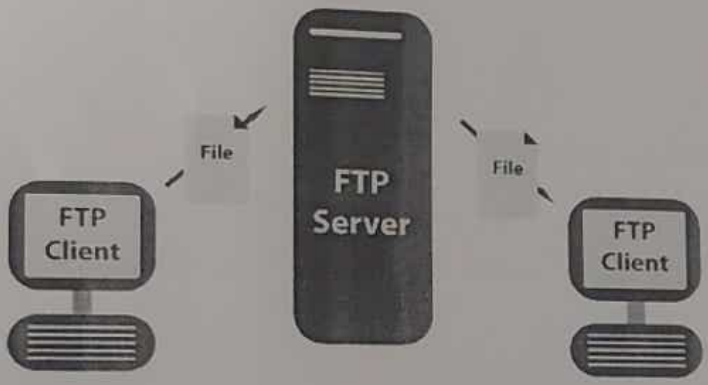


Figure 15: File Transfer Protocol

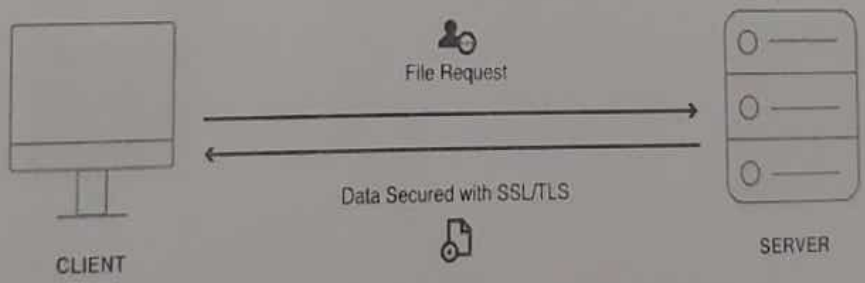


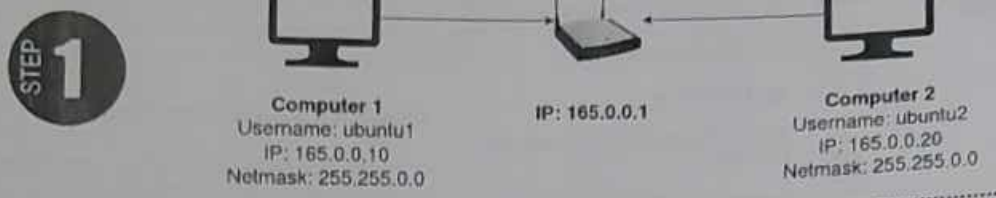
Figure 16: How Does File Transfer Protocol Work?

Required Components

- Computers or Laptops - 02
- Manageable Router / Switch - 01
- LAN Cables - 02

Network Diagram & Procedure

1. Connect the Network



Computer 1
Username: ubuntu1
IP: 165.0.0.10
Netmask: 255.255.0.0

IP: 165.0.0.1

Computer 2
Username: ubuntu2
IP: 165.0.0.20
Netmask: 255.255.0.0

2. System Preparation

Connect the System to Internet with Wired or Wireless Connectivity	Connect the System to Internet with Wired or Wireless Connectivity
Update the System using the command **sudo apt-get update**	Update the System using the command **sudo apt-get update**
Install the ProFTPD Software using the command **sudo apt-get install proftpd**	Install the FTP Software using the command **sudo apt-get install ftp**

Disconnect the Internet Connectivity	Disconnect the Internet Connectivity
Open the Terminal (Ctrl + Alt + T)	Open the Terminal (Ctrl + Alt + T)
Type the command - **sudo ifconfig**	Type the command - **sudo ifconfig**
Observe the ethernet interface. Find the ID of the ethernet interface card	Observe the ethernet interface. Find the ID of the ethernet interface card
Type the command - **sudo systemctl stop NetworkManager** to stop the networking services	Type the command - **sudo systemctl stop NetworkManager** to stop the networking services
Type the following command to set the IP address - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**	Type the following command to set the IP address - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**
Use the command - **sudo ifconfig** to check the IP address is set to the ethernet interface	Use the command - **sudo ifconfig** to check the IP address is set to the ethernet interface
Type the command **ping <ip_address>** of Computer 2	Type the command **ping <ip_address>** of Computer 1

Figure 17: Network Diagram & Procedure

4. File Transfer using PUT

Change the Working Directory to a location to Copy the Files

cd <location_of_directory>

Initiate the File Transfer using

ftp 165.0.0.10

Enter the Computer 1 - Username and Password

Username: ubuntu1
Password: ******

Transfer the File using the command

put <filename>

Terminate the FTP Connection using

bye

Verify the copied file in the Current Working Directory using the command

ls

The student will be able to learn

1. to configure two machines as client and server and demonstrate file transfer protocol

2. to update the system and install the required packages for accomplishing file transfer

5. File Transfer using GET

Change the Working Directory to a location to Copy the Files

cd <location_of_directory>

Initiate the File Transfer using

ftp 165.0.0.20

Enter the Computer 1 - Username and Password

Username: ubuntu2
Password: ******

Transfer the File using the command

get <filename>

Terminate the FTP Connection using

bye

Verify the copied file in the Current Working Directory using the command

ls

The student will be able to implement

1. File Transfer Protocol (FTP) with multiple commands like put, get and explore the other commands

2. Locate the Directory on a remote system

3. Accomplish fetch and copy files with appropriate permissions

Figure 18: Network Diagram & Procedure

```
rvcemca@ubuntu22:~$ sudo apt update                                    Step 2
Ign:1 http://dl.google.com/linux/chrome-remote-desktop/deb stable InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://dl.google.com/linux/chrome-remote-desktop/deb stable Release
Hit:6 http://in.archive.ubuntu.com/linux/chrome/deb stable InRelease
Hit:7 http://archive.canonical.com/ubuntu jammy-updates InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
rvcemca@ubuntu22:~$ sudo apt-get install proftpd                       Step 3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'proftpd-core' instead of 'proftpd'
proftpd-core is already the newest version (1.3.7c+dfsg-1build1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
rvcemca@ubuntu22:~$
```

Figure 19: Update and Installation of ProFTPD on Server

```
deepika@debian:~$ sudo apt-get update                                  Step 4
Ign:1 http://dl.google.com/linux/chrome-remote-desktop/deb stable InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 https://apt.grafana.com stable InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 https://brave-browser-apt-release.s3.brave.com stable InRelease
Hit:6 http://deb.debian.org/debian bullseye InRelease
Hit:7 http://dl.google.com/linux/chrome-remote-desktop/deb stable Release
Hit:8 http://deb.debian.org/debian bullseye-updates InRelease
Hit:9 http://ftp.de.debian.org/debian bullseye InRelease
Reading package lists... Done
deepika@debian:~$ sudo apt-get install ftp                             Step 5
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ftp is already the newest version (0.17-34.1.1).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
deepika@debian:~$
```

Figure 20: Update and Installation of FTP on Client

Step 1: Make a network diagram as shown in Figure 17 with necessary information such as **Username, IP and Netmask** for all the machines.

Step 2: Connect the Server System to Internet using Ethernet or Wi-Fi connection and update the system.

Step 3: Install **ProFTPD** on the Server machine using the command - **sudo apt-get install proftpd** as shown in Figure 19

Step 4: Connect the Client System to Internet using Ethernet or Wi-Fi connection and update the system.

Step 5: Install **FTP** on the Client machine using the command - **sudo apt-get install ftp** as represented in Figure 20

```
rvcenca@ubuntu22:~$ sudo ifconfig                                    Step 6
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::f99a:edd2:83d:2acc  prefixlen 64  scopeid 0x20<link>
        ether c8:5a:cf:a0:7c:28  txqueuelen 1000  (Ethernet)
        RX packets 283  bytes 76457 (76.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 741  bytes 146541 (146.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 10650  bytes 1098828 (1.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10650  bytes 1098828 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

rvcenca@ubuntu22:~$ sudo systemctl stop NetworkManager               Step 7
rvcenca@ubuntu22:~$ sudo ifconfig enp2s0 165.0.0.10 netmask 255.255.0.0 up   Step 8
rvcenca@ubuntu22:~$ sudo ifconfig                                    Step 9
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 165.0.0.10  netmask 255.255.0.0  broadcast 165.0.255.255
        inet6 fe80::f99a:edd2:83d:2acc  prefixlen 64  scopeid 0x20<link>
        ether c8:5a:cf:a0:7c:28  txqueuelen 1000  (Ethernet)
        RX packets 283  bytes 76457 (76.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 772  bytes 152635 (152.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 21: Setting the IP, Terminating NetworkManager, Assigning a new IP - Server

Step 6: Disconnect the Server Machine from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Server using the command - **sudo ifconfig**

Step 7: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services.

Step 8: Set the IP address using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**.

Step 9: Check the IP Address using the command - **sudo ifconfig** as given in Figure 21

Step 10: Disconnect the Client Machine from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Client using the command - **sudo ifconfig**

Step 11: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services.

Step 12: Set the IP address using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**.

Step 13: Check the IP Address using the command - **sudo ifconfig** as given in Figure 22

Step 14: Navigate into the location of the file to be copied. Now **'cd'** to the directory where you have the file you want to upload via ftp. For example, **/usr/share/common-licenses**. Use the

```
deepika@debian:~$ sudo ifconfig                                    Step 10
enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 3c:97:0e:e4:6e:cf  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 106  bytes 18908 (18.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf2500000-f2520000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 370  bytes 32179 (31.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 370  bytes 32179 (31.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

deepika@debian:~$ sudo systemctl stop NetworkManager          Step 11
deepika@debian:~$ sudo ifconfig enp0s25 165.0.0.20 netmask 255.255.0.0 up   Step 12
deepika@debian:~$ sudo ifconfig                                    Step 13
enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 165.0.0.20  netmask 255.255.0.0  broadcast 165.0.255.255
        ether 3c:97:0e:e4:6e:cf  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 116  bytes 20042 (19.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf2500000-f2520000
```

Figure 22: Setting the IP, Terminating NetworkManager, Assigning a new IP - Client

command **cd /usr/share/common-licenses**. A file called: **GPL-3** is present. Upload this file.

Step 15: Run the following to ftp to the computer using **ftp <ip-of-proftpd-computer>**. Type the **Name** and **Password** when prompted

Step 16: Run the command **ls** to list the files present in the current directory

Step 17: Use the command **put GPL-3** to transfer the file.

Step 18: Run the command **ls** to list the files present in the current directory and check whether GPL-3 is present with the location

Step 19: Create a file **sample.txt**. Use the command **get sample.txt** to receive the file.

Step 20: Run the command **ls** to list the files present in the current directory and check whether **ls** is present with the location

Expected Output

- The students will be able to configure two machines as client and server and demonstrate file transfer protocol
- The student will be able to be able to understand and implement file transfer protocol

```
deepika@debian:~$ cd /usr/share/common-licenses/                    Step 14
deepika@debian:/usr/share/common-licenses$ ftp 165.0.0.10           Step 15
Connected to 165.0.0.10.
220 ProFTPD Server (Debian) [::ffff:165.0.0.10]
Name (165.0.0.10:deepika): rvcemca
331 Password required for rvcemca
Password:
230 User rvcemca logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls                                                             Step 16
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 26 09:46 Desktop
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Documents
drwxr-xr-x   3 rvcemca  rvcemca      4096 Sep  1 14:29 Downloads
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Music
drwxr-xr-x   3 rvcemca  rvcemca      4096 Dec 26 10:45 Pictures
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Public
drwx------   7 rvcemca  rvcemca      4096 Dec 26 09:58 snap
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Templates
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Videos
226 Transfer complete
```

Figure 23: Initiating FTP to another System

```
ftp> put GPL-3                                                      Step 17
local: GPL-3 remote: GPL-3
200 PORT command successful
150 Opening BINARY mode data connection for GPL-3
226 Transfer complete
35147 bytes sent in 0.00 secs (165.1172 MB/s)
ftp> ls                                                             Step 18
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 26 09:46 Desktop
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Documents
drwxr-xr-x   3 rvcemca  rvcemca      4096 Sep  1 14:29 Downloads
-rw-r--r--   1 rvcemca  rvcemca     35147 Dec 26 11:37 GPL-3
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Music
drwxr-xr-x   3 rvcemca  rvcemca      4096 Dec 26 10:45 Pictures
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Public
drwx------   7 rvcemca  rvcemca      4096 Dec 26 09:58 snap
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Templates
drwxr-xr-x   2 rvcemca  rvcemca      4096 Dec 24  2021 Videos
226 Transfer complete
```

Figure 24: Transfer a File using **put** Command

```
ftp> get sample.txt                                                 Step 19
local: sample.txt remote: sample.txt
200 PORT command successful
150 Opening BINARY mode data connection for sample.txt (52 bytes)
226 Transfer complete
52 bytes received in 0.00 secs (246.5109 kB/s)
ftp> exit
221 Goodbye.
deepika@debian:/usr/share/common-licenses$ ls                       Step 20
Apache-2.0  BSD       GFDL      GFDL-1.3  GPL-1  GPL-3  LGPL-2    LGPL-3   MPL-2.0
Artistic    CC0-1.0   GFDL-1.2  GPL       GPL-2  LGPL   LGPL-2.1  MPL-1.1  sample.txt
deepika@debian:/usr/share/common-licenses$
```

Figure 25: Receive a File using **get** Command

Program - 3

Demonstrate secured file transfer and computing over wired Network and wireless Network with SSH key based computing and SCP

Description

Design a network topology with two nodes - PC1 and PC2. The nodes - PC1 and PC2 are connected to a 8 port switch in centre as shown in Figure 28. Set the node PC1 as a client and PC2 as a server.

Secure Shell - SSH

SSH (Secure SHell) is a cryptographic network protocol for operating network services securely over an unsecured network. The best known example content application is for remote login to computer systems by users.

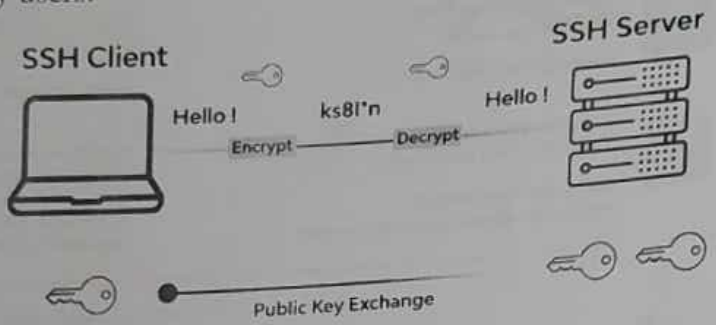


Figure 26: Secure Shell Protocol

Secure Copy - SCP

SCP (Secure CoPy) is a command line utility that allows you to securely copy files and directories between two locations. With scp, a file or directory can be copied from a local system to a remote system.

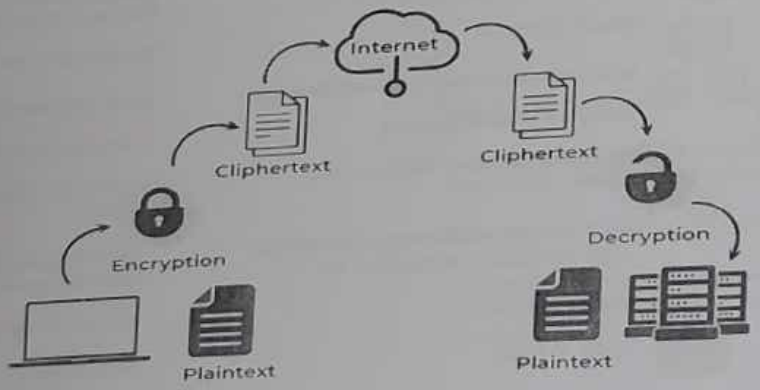


Figure 27: Secure Copy Protocol

Required Components

- Computers or Laptops - 02
- Manageable Router / Switch - 01
- LAN Cables - 02

Network Diagram & Procedure

1. Connect the Network

Computer 1
Username: ubuntu1
IP: 192.168.0.10
Netmask: 255.255.255.0

IP: 192.168.0.1

Computer 2
Username: ubuntu2
IP: 192.168.0.20
Netmask: 255.255.255.0

2. System Preparation

Connect the System to Internet with Wired or Wireless Connectivity	Connect the System to Internet with Wired or Wireless Connectivity
Update the System using the command **sudo apt-get update**	Update the System using the command **sudo apt-get update**
Install the OpenSSH Server and Client Softwares using the command **sudo apt-get install openssh-client openssh-server**	Install the OpenSSH Server and Client Softwares using the command **sudo apt-get install openssh-client openssh-server**

Disconnect the Internet Connectivity	Disconnect the Internet Connectivity
Open the Terminal (Ctrl + Alt + T) Type the command - **sudo ifconfig**	Open the Terminal (Ctrl + Alt + T) Type the command - **sudo ifconfig**
Observe the ethernet interface. Find the ID of the ethernet interface card Type the command - **sudo systemctl stop NetworkManager** to stop the networking services	Observe the ethernet interface. Find the ID of the ethernet interface card Type the command - **sudo systemctl stop NetworkManager** to stop the networking services
Type the following command to set the IP address - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up** Use the command - **sudo ifconfig** to check the IP address is set to the ethernet interface	Type the following command to set the IP address - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up** Use the command - **sudo ifconfig** to check the IP address is set to the ethernet interface
Type the command **ping <ip_address>** of Computer 2	Type the command **ping <ip_address>** of Computer 1

Figure 28: Network Diagram & Procedure

4. Generate the Keys and Secure Copy

The student will be able to learn

1. *to configure two machines as client and server and demonstrate secure login and secure copy*
2. *to update the system and install the required packages for accomplishing secure login*

Generate SSH Keys with the command

ssh-keygen

Enter the filename and passphrase.

The filename is set to ***key*** in this example.

Public Key will be

~/.ssh/key.pub

Private Key will be

~/.ssh/key

Copy the file to the other system using the command

scp ~/key.pub pi@192.168.0.10:/home /username

5. Concatenation of File and Secure Login using Keys

Concatenate the public key to the Authorized Keys

cat key.pub >> ~/.ssh/authorized_keys

Verify the concatenated key by viewing the file

cat .ssh/authorized_keys

Login into the other system using

ssh -i ~/key pi@192.168.0.10

The student will be able to implement

1. *Secure SHell (SSH) key generation and logging into remote machine using Secure Login*
2. *Locate the Directory on a remote system*
3. *Accomplish key transfer, copy Public Key using secure copy into authorized keys and login using Private Key*

Figure 29: Network Diagram & Procedure

Install SSH & Make a Copy of the Original File

sudo apt-get install openssh-server openssh-client
sudo cp /etc/ssh/sshd_config /etc/ssh/ssh_config.original_copy

Process for performing Secure Shell

Step 1: Make a network diagram as shown in Figure 28 with necessary information such as **Username, IP and Netmask** for all the machines.

Step 2: Connect the Server System to Internet using Ethernet or Wi-Fi connection and update the system.

Step 3: Install **OpenSSH** on the Server and Client machines using the command - **sudo apt-get install openssh server openssh client**

Step 4: Disconnect the Server Machine from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Server using the command - **sudo ifconfig**

Step 5: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services.

Step 6: Set the IP address using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**.

Step 7: Check the IP Address using the command - **sudo ifconfig**

Step 8: On Computer 1, run: **ssh-keygen**. Save the keys as with a filename. The file is saved as ***key*** for explanation process. Refer Figure 30.

Step 9: Run the command **ls** to verify if the keys have been generated. The public key is: **/.ssh/key.pub** and the private key is: **/.ssh/key**.

Note

Never share the private key with anyone. Public key can be distributed widely.

Step 10: On Server, run: **scp ~/key.pub deepika@192.168.0.20:~/Downloads** to initiate **Secure Copy Protocol** to transfer the file securely to Client.

```
rvcemca@ubuntu22:~$ ssh-keygen                                    Step 8
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rvcemca/.ssh/id_rsa): key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key
Your public key has been saved in key.pub
The key fingerprint is:
SHA256:OZKzVKJHR9inn4TPbfhSYy+SBnBIvWWllR+r4aHmTsg rvcemca@ubuntu22
The key's randomart image is:
+---[RSA 3072]----+
|        o.       |
|       ... .  .  |
|       o o+  + . |
|      o =o..+ . o|
|     . * S=o+o o |
|      o * =*+*+  |
|       . E **+o  |
|         .*+oo . |
|          +=o .  |
+----[SHA256]-----+
rvcemca@ubuntu22:~$ ls                                            Step 9
Desktop  Documents  Downloads  key  key.pub  Music  Pictures  Public  snap  Templates  Videos
rvcemca@ubuntu22:~$ sudo scp ~/key.pub deepika@192.168.0.20:~/Downloads   Step 10
deepika@192.168.0.20's password:
key.pub
```

Figure 30: Creation of Secure Shell Keys

Step 11: On Computer 2, run **cat key.pub » ~/.ssh/authorized_keys** and **cat .ssh/authorized_keys**

```
deepika@debian:~/Downloads$ cat key.pub >> ~/.ssh/authorized_keys     Step 11
deepika@debian:~/Downloads$ cat ~/.ssh/authorized_keys                Step 12
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDEz6g+/EwrXFojuX58LbXcetQB1WYOKP0F9OdfFVxmuwD3vm117Ah6dnwVL JVGHGd1xovXO
Gmp589bpTDJ4alnnAR0FMfv1umMx5/QCRCzHUQrBQ8t7UR1FVkzPa9OH5GNETOkBQW83crC8MdbZrWR0K5qx/clMTSBXP6Utaczb1QD5fPoOa
h5LAKkJC3Dj15CRAeIm312+x1hhw220VB5L8ak7MCqxO1KKYwcINYtXV6j9ai/R49Hrtn+OMYc1PaPQ/fPuTOHyYg2vRPqGJGqEaKHrvs7B6f
dqvz3N1CHEDQfZHJ1ce4aWBUgsP/fV1u/F3G9s3b/ZhcbpI9d1t+Phq8zqus8Im2kv4bas/ZB3AW7rmzFRJ5a4BxQS3yVlboXbletD/U2lUrt
qI+NeDjtJQ/fgYVyUIpmB/tojmVTqFHDCzp6egNJ1rAjZgclke6Q8eh8gmyAD0EWXQ7ncXGlX/CtdZYOlFzWcNS7ufyPVHfyLKg8Amz3MGKzY
OCk2+c= rvcemca@ubuntu22
deepika@debian:~/Downloads$
```

Figure 31: Concatenate Secure Shell Keys

Step 12: Verify if **key.pub** is concatenated to **~/.ssh/authorized_keys**

Step 13: Login into the Computer 2 with the command **ssh -i ~/key deepika@192.168.0.20**

```
rvcemca@ubuntu22:~$ ssh -i ~/key deepika@192.168.0.20             Step 13
Enter passphrase for key '/home/rvcemca/key':
Linux debian 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 26 13:22:27 2022 from 192.168.0.10
deepika@debian:~$
```

Figure 32: Login using Secure Shell Keys

Program - 4

Demonstrate to calculate IP addresses using ipcalc

Description

Design a network topology with one node - PC1 to understand and demonstrate to calculate IP addresses using ipcalc

Ipcalc

Ipcalc is an IP address and netmask and provides the resulting broadcast, network, Cisco wildcard mask, and host range. The administrators can create subnets and supernets by specifying a second netmask. It's also meant to be a teaching tool. Therefore, the subnetting results are presented as simple binary values.

Features:

- Formats for multiple addresses and netmask output (dotted quad, hex, number of bits) and Bitmaps of various types are output. A user-defined number of extra networks are output.
- Multiple networks can be accessed using the command line with Hostname DNS resolutions.

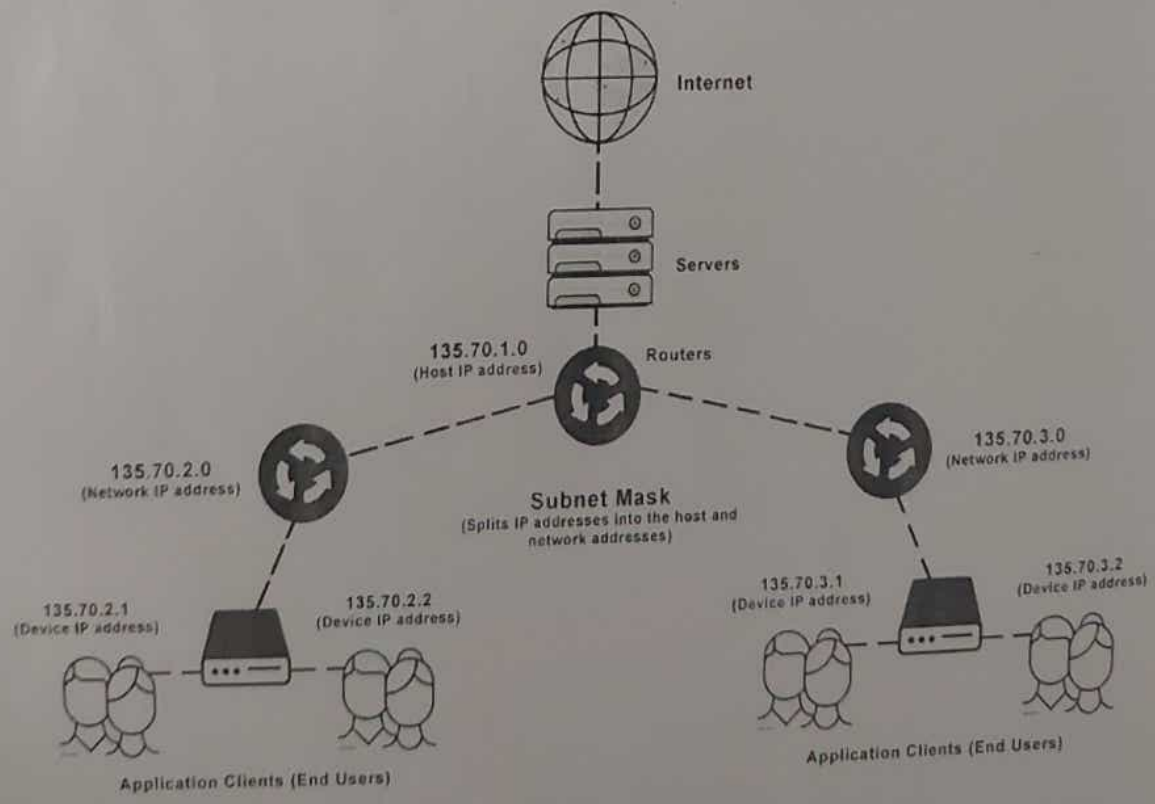


Figure 33: Representation of **ipcalc**

Required Components

- puters or Laptops - 01

Step 1: Connect the Server System to Internet using Ethernet or Wi-Fi connection and update the system.

Step 2: Install Ipcalc on the system using the command **sudo apt-get install ipcalc**.

```
deepika@debian:~$ sudo apt-get install ipcalc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ipcalc is already the newest version (0.42-2).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
```

Figure 35: Installation of Ipcalc

Step 3: Disconnect the Internet from the system. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Server using the command - **sudo ifconfig**.

Step 4: Set the IP address using - **sudo ifconfig <ethernet_interface_id <ip> netmask <netmask> up.**

Step 5: Use ipcalc with the assigned IP address to get information about the network address **ipcalc <ip>**.

```
deepika@debian:~$ ipcalc 192.168.0.20
Address:   192.168.0.20          11000000.10101000.00000000. 00010100
Netmask:   255.255.255.0 = 24    11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255             00000000.00000000.00000000. 11111111
=>
Network:   192.168.0.0/24        11000000.10101000.00000000. 00000000
HostMin:   192.168.0.1           11000000.10101000.00000000. 00000001
HostMax:   192.168.0.254         11000000.10101000.00000000. 11111110
Broadcast: 192.168.0.255         11000000.10101000.00000000. 11111111
Hosts/Net: 254                    Class C, Private Internet

deepika@debian:~$ ipcalc 192.168.0.10
Address:   192.168.0.10          11000000.10101000.00000000. 00001010
Netmask:   255.255.255.0 = 24    11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255             00000000.00000000.00000000. 11111111
=>
Network:   192.168.0.0/24        11000000.10101000.00000000. 00000000
HostMin:   192.168.0.1           11000000.10101000.00000000. 00000001
HostMax:   192.168.0.254         11000000.10101000.00000000. 11111110
Broadcast: 192.168.0.255         11000000.10101000.00000000. 11111111
Hosts/Net: 254                    Class C, Private Internet
```

```
deepika@debian:~$ ipcalc 165.0.1.15
Address:   165.0.1.15           10100101.00000000.00000001. 00001111
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   165.0.1.0/24         10100101.00000000.00000001. 00000000
HostMin:   165.0.1.1            10100101.00000000.00000001. 00000001
HostMax:   165.0.1.254          10100101.00000000.00000001. 11111110
Broadcast: 165.0.1.255          10100101.00000000.00000001. 11111111
Hosts/Net: 254                   Class B

deepika@debian:~$ ipcalc 176.0.0.40
Address:   176.0.0.40           10110000.00000000.00000000. 00101000
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   176.0.0.0/24         10110000.00000000.00000000. 00000000
HostMin:   176.0.0.1            10110000.00000000.00000000. 00000001
HostMax:   176.0.0.254          10110000.00000000.00000000. 11111110
Broadcast: 176.0.0.255          10110000.00000000.00000000. 11111111
Hosts/Net: 254                   Class B
```

Figure 36: Using Ipcalc with IP Address

Step 6:Calculate a subnet for an IP address using the command

ipcalc <ip>/<**subnet**> **-s <hosts>**

Step 6

```
deepika@debian:~$ ipcalc 165.0.1.15/16 -s 15 15
Address:   165.0.1.15           10100101.00000000. 00000001.00001111
Netmask:   255.255.0.0 = 16     11111111.11111111. 00000000.00000000
Wildcard:  0.0.255.255          00000000.00000000. 11111111.11111111
=>
Network:   165.0.0.0/16         10100101.00000000. 00000000.00000000
HostMin:   165.0.0.1            10100101.00000000. 00000000.00000001
HostMax:   165.0.255.254        10100101.00000000. 11111111.11111110
Broadcast: 165.0.255.255        10100101.00000000. 11111111.11111111
Hosts/Net: 65534                 Class B

1. Requested size: 15 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   165.0.0.0/27         10100101.00000000.00000000.000 00000
HostMin:   165.0.0.1            10100101.00000000.00000000.000 00001
HostMax:   165.0.0.30           10100101.00000000.00000000.000 11110
Broadcast: 165.0.0.31           10100101.00000000.00000000.000 11111
Hosts/Net: 30                    Class B

2. Requested size: 15 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   165.0.0.32/27        10100101.00000000.00000000.001 00000
HostMin:   165.0.0.33           10100101.00000000.00000000.001 00001
HostMax:   165.0.0.62           10100101.00000000.00000000.001 11110
Broadcast: 165.0.0.63           10100101.00000000.00000000.001 11111
Hosts/Net: 30                    Class B
```

```
Needed size:  64 addresses.
Used network: 165.0.0.0/26
Unused:
165.0.0.64/26
165.0.0.128/25
165.0.1.0/24
165.0.2.0/23
165.0.4.0/22
165.0.8.0/21
165.0.16.0/20
165.0.32.0/19
165.0.64.0/18
165.0.128.0/17
```

Figure 37: Subnetting using Ipcalc

Expected Output

- The student will learn to validate IP address
- The student will be able to show calculated broadcast address and network address or prefix

Sl. No	Rubrics for Practice Session		Total Marks	Obtained Marks
1	**Conduction & Execution**	Understanding of Concepts & logic	**02**	1.5
2		Approach Towards the Problem	**02**	02
3		Conduction and Execution: Input & Output for allpossible cases	**03**	03
4	**Viva Voce**	Program Analysis & Applications	**02**	1.5
5		Communication & Confidence Level	**01**	0.5
Total			**10**	8.5

Program - 5

Build DHCP server using dns-masq with and without MAC binding with IPV4 and IPV6

Description

Design a network topology with two nodes PC1 and PC2. The node PC1 and PC2 is connected to a 8 port switch in the center as shown in Figure 39. Assign IP address to each node and implement Dynamic Host Configuration Protocol (DHCP).

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a network management protocol used on UDP/IP networks whereby a DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so they can communicate with other IP networks. The representation of DHCP is provided in Figure 38

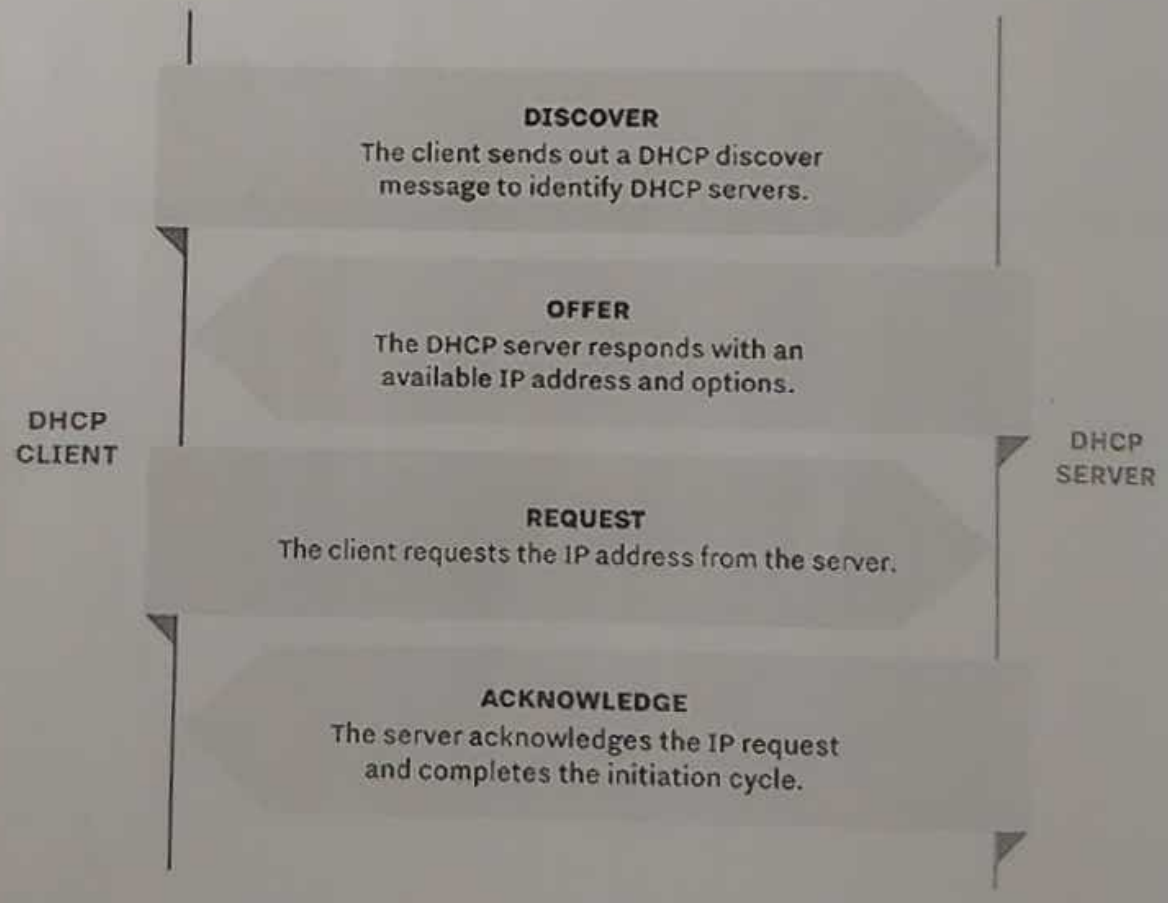


Figure 38: Representation of DHCP Handshake

Required Components

- Computers or Laptops - 02
- Manageable Router / Switch - 01
- LAN Cables - 02

Network Diagram

1. Connect the Network

Computer 1
Username: ubuntu1
IP: 175.50.0.10
Netmask: 255.255.0.0

IP: 175.50.0.1

Computer 2
Username: ubuntu2
IP: 175.50.0.20
Netmask: 255.255.0.0

2. System Preparation

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the dnsmasq Software using the command

sudo apt-get install dnsmasq

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the dnsmasq Software using the command

sudo apt-get install dnsmasq

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Terminate the services of dnsmasq

sudo systemctl stop dnsmasq

Verify the IP address of the interface

sudo ifconfig <interface name>

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Terminate the services of dnsmasq

sudo systemctl stop dnsmasq

Verify the IP address of the interface

sudo ifconfig <interface name>

4. Method 1 - Via Command-Line

Type the following command to set the IP address -

sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up

Reinitiate the dnsmasq services and specify the range with log using the command

sudo dnsmasq -I enp0s25 -I lo -a <ip> -dhcp-range=<ip_start_range>, <ip_end_range> -d -u root -log-dhcp -q

Fetch an IP Address in the specified range

sudo dhclient -v

5. Method 2 - Via Configuration File

Configure dnsmasq services using

sudo nano /etc/dnsmasq.conr

Edit the configuration file

 # DNS configuration

```
port=53
domain-needed
bogus-priv
strict-order
expand-hosts
dhcp-range=<ip_start_range>,<ip_end_range>
dhcp-option=option:router, <ip_address.1>
dhcp-option=option:netmask,<subnet>
```

Re-initiate dnsmasq services

sudo systemctl start dnsmasq

Verify the status of dnsmasq

sudo systemctl status dnsmasq

Fetch an IP Address in the specified range

sudo dhclient -v

Verify the IP Address with

sudo ifconfig

Attention:

In the case of using a router, disable the DHCP Server services by entering the router's default IP address (**http://192.168.0.1**) in the browser and terminate the DHCP Server by setting it to **None**.

Step 1: Connect the Client and Server Systems to Internet using Ethernet or Wi-Fi connection and update the system.

Step 2: Install **dnsmasq** on the machines using the command - **sudo apt-get install dnsmasq**

Step 3: Disconnect the machines from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the machines using the command - **sudo ifconfig**

Step 4: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services on the machines.

Step 5: Stop the **dnsmasq** services using - **sudo systemctl stop dnsmasq**.

Step 6: Verify the IP address with the interface name using the command **sudo ifconfig <interface_name>**.

```
deepika@debian:~$ sudo apt-get install dnsmasq                    Step 2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dnsmasq is already the newest version (2.85-1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
deepika@debian:~$ sudo systemctl stop NetworkManager              Step 4
deepika@debian:~$ sudo systemctl stop dnsmasq                     Step 5
deepika@debian:~$ sudo ifconfig enp0s25                           Step 6
enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::cd35:af66:8aa4:2b9b  prefixlen 64  scopeid 0x20<link>
        ether 3c:97:0e:e4:6e:cf  txqueuelen 1000  (Ethernet)
        RX packets 34  bytes 2272 (2.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34  bytes 6264 (6.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf2500000-f2520000

deepika@debian:~$ sudo ifconfig enp0s25 120.0.0.10 netmask 255.0.0.0 up   Step 7
deepika@debian:~$ sudo dnsmasq -i enp0s25 -I lo -a 120.0.0.10 --dhcp-range=120
-log-dhcp -q                                                      Step 8
dnsmasq: started, version 2.85 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHC
 auth cryptohash DNSSEC loop-detect inotify dumpfile
dnsmasq-dhcp: DHCP, IP range 120.0.0.1 -- 120.0.0.100, lease time 1h
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 172.16.2.35#53
dnsmasq: read /etc/hosts - 7 addresses
```

Figure 40: DHCP Server using dnsmasq with Command-Line Method - Server Side

Method 1: Via Command-line

Step 7: Set the IP address on the Server machine using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**

Make a Copy of the Original File
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.bck

Step 8: Reinitiate the **dnsmasq** services and specify the range with log using the command **sudo dnsmasq -i enp0s25 -I lo -a <ip> –dhcp-range=<ip_start_range>,<ip_end_range> -d -u root –log-dhcp -q**. Refer Figure No. 40

Step 9: On the client system, run the command **sudo dhclient -v**

Step 10: Verify the IP address with the interface name using the command **sudo ifconfig <interface_name>**. The IP address obtained with **dhclient** will be an address be one inside the range.

sudo systemctl stop systemd-resolve
-resolved

```
rvcemca@ubuntu22:~$ sudo systemctl stop NetworkManager
rvcemca@ubuntu22:~$ sudo dhclient -v
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlp0s20f3/84:14:4d:09:b5:64
Sending on   LPF/wlp0s20f3/84:14:4d:09:b5:64
Listening on LPF/enp2s0/c8:5a:cf:a0:7c:28
Sending on   LPF/enp2s0/c8:5a:cf:a0:7c:28
Sending on   Socket/fallback
DHCPDISCOVER on wlp0s20f3 to 255.255.255.255 port 67 interval 3 (xid=0xa9f8fb74)
DHCPREQUEST for 120.0.0.36 on enp2s0 to 255.255.255.255 port 67 (xid=0x53446936)
DHCPACK of 120.0.0.36 from 120.0.0.10 (xid=0x36694453)
RTNETLINK answers: File exists
```

Step 4

Step 9

Figure 41: DHCP Server using dnsmasq with Command-Line Method - Client Side

Method 2: Via Configuration File

Step 11: Execute till **Step 6** and configure **dnsmasq** services using - **sudo nano /etc/dnsmasq.conf**

Step 12: Edit the configuration file as shown in the Figure 42.

```
 GNU nano 5.4                                  /etc/dnsmasq.conf
port=53                                                    Step 12
domain-needed
bogus-priv
strict-order
expand-hosts
dhcp-range=175.50.0.100,175.50.0.200,24h
dhcp-option=option:router,175.50.0.1
dhcp-option=option:netmask,255.255.0.0
```

Figure 42: DHCP Server using dnsmasq with Command-Line Method - Server Side

Step 13: Reinitiate **dnsmasq** services using - **sudo systemctl start dnsmasq** and no errors should appear.

Step 14: Verify **dnsmasq** status using - **sudo systemctl status dnsmasq**
The **dnsmasq** service must be **loaded** and **active(running)**

Step 15: On the Client system, run the command **sudo dhclient <interface_name>** or reboot the system.

Step 16: Verify the interface with the command **sudo ifconfig <interface_name>**
The IP address obtained with **dhclient** will be an address be one within the range.

```
                                                           Step 4
rvcemca@ubuntu22:~$ sudo systemctl stop NetworkManager
                                                           Step 14
[sudo] password for rvcemca:
rvcemca@ubuntu22:~$ sudo dhclient -v
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlp0s20f3/84:14:4d:09:b5:64
Sending on   LPF/wlp0s20f3/84:14:4d:09:b5:64
Listening on LPF/enp2s0/c8:5a:cf:a0:7c:28
Sending on   LPF/enp2s0/c8:5a:cf:a0:7c:28
Sending on   Socket/fallback
DHCPDISCOVER on wlp0s20f3 to 255.255.255.255 port 67 interval 3 (xid=0x5e13c93c)
DHCPREQUEST for 175.50.0.193 on enp2s0 to 255.255.255.255 port 67 (xid=0x5c4755b3)
DHCPACK of 175.50.0.193 from 175.50.0.10 (xid=0xb355475c)
RTNETLINK answers: File exists
bound to 175.50.0.193 -- renewal in 32674 seconds.
                                                           Step 15
rvcemca@ubuntu22:~$ sudo ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 175.50.0.193  netmask 255.255.0.0  broadcast 175.50.255.255
        ether c8:5a:cf:a0:7c:28  txqueuelen 1000  (Ethernet)
```

Figure 43: DHCP Server using dnsmasq with Configuration File Method - Client Side

Expected Output

Program - 6

Build DNS server for resolving the names and IP addresses

Description

Design a network topology with a single node PC1 Figure 49. Assign IP address to the node and implement Domain Name System (DNS).

Domain Name System (DNS)

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. The basic representation of DNS is shown in Figure 44.

Every website has an IP address. Two types of addresses are currently in use:

- IPv4—a four-byte address: four numbers ranging from 0 to 255. The format looks like this: 000.111.222.123.
- IPv6—a more modern type of IP, a 16-byte address. There are several formats to record IPv6 addresses. The most widespread one is 8 groups with 4 symbols in each (eight four-digit hexadecimal numbers), divided by colons, which might look like this: 1234:abcd:1b4d:000a:987c:5555:a2d8:bcd6.

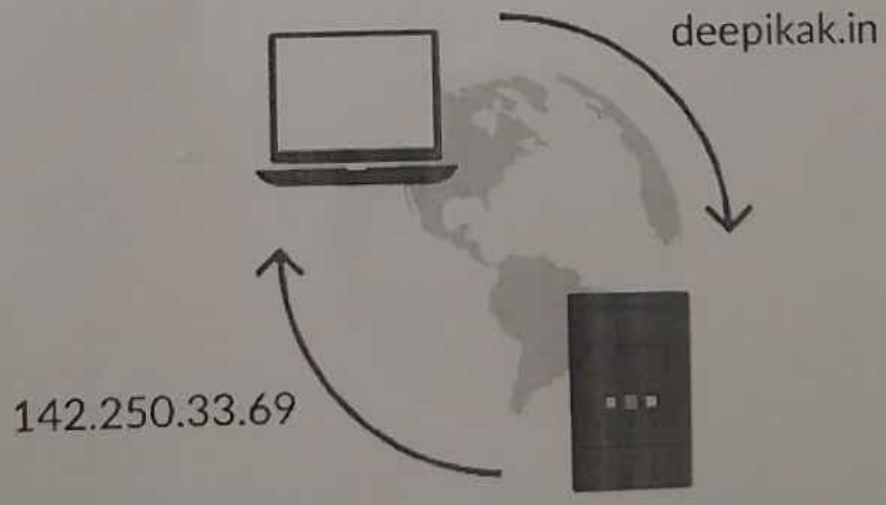


Figure 44: Domain Name System

Required Components

- Computers or Laptops - 01

20MCA13

Network Diagram & Procedure

1. Connect the Network

STEP 1

Computer 1
Username: ubuntu1
IP: 192.168.0.10
Netmask: 255.255.255.0

2. System Preparation

STEP 2

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the dnsmasq and dnsutils Software using the command

sudo apt-get install dnsmasq
sudo apt-get install dnsutils

3. Setting up the IP

STEP 3

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command :

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Type the following command to move the original configuration file to another location or rename it in the same location

sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

Create a new configuration file for dnsmasq

sudo nano /etc/dnsmasq.conf

Edit the dnsutils configuration file

sudo nano /etc/hosts

Restart the dnsmasq services using

sudo systemctl restart dnsmasq

Fetch the IP specified for the domain in the hosts file

dig a <domain_name> @localhost

 # DNS configuration

```
port=53
domain-needed
bogus-priv
strict-order
expand-hosts
domain=deepikak.in
```

 # Host addresses

```
127.0.0.1  localhost
127.0.1.1  deepika-debian
142.250.33.69 deepikak.in
142.250.33.78 cn.manual.deepikak.in
```

Figure 45: Network Diagram & Procedure

```
deepika@debian:~$ sudo apt-get install dnsmasq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dnsmasq is already the newest version (2.85-1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
deepika@debian:~$ sudo apt-get install dnsutils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dnsutils is already the newest version (1:9.16.37-1~deb11u1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
deepika@debian:~$ sudo systemctl stop NetworkManager
deepika@debian:~$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
deepika@debian:~$ sudo nano /etc/dnsmasq.conf
deepika@debian:~$ sudo nano /etc/hosts
deepika@debian:~$ sudo systemctl restart dnsmasq
deepika@debian:~$ dig a deepikak.in @localhost

; <<>> DiG 9.16.37-Debian <<>> a deepikak.in @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59430
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; ANSWER SECTION:
deepikak.in.            0       IN      A       142.250.33.69
```

Figure 46: Procedure to implement DNS Server

Step 1: Connect the system to Internet using Ethernet or Wi-Fi connection and update the system.

Step 2: Install **dnsmasq** on the system using the command **sudo apt-get install dnsmasq**.

Step 3: Install **dnsutils** on the system using the command **sudo apt-get install dnsutils**.

Step 4: Disconnect the Internet from the system. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the machine using the command - **sudo ifconfig**.

Step 5: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services on both the machine.

Step 6: Move the original copy of the original configuration to another location with **sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig**

```
GNU nano 5.4                    /etc/dnsmasq.conf *
# DNS configuration
    port=53

    domain-needed
    bogus-priv
    strict-order

    expand-hosts
    domain=deepikak.in
```

Step 8

```
GNU nano 5.4                    /etc/hosts
# Host addresses
127.0.0.1  localhost
127.0.1.1  deepika-debian
142.250.33.69 deepikak.in
142.250.33.78 cn.manual.deepikak.in

::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Step 10

Figure 47: Configuration Files of **dnsmasq** and **hosts**

Step 7: Configure the **dnsmasq** services with the command **sudo nano /etc/dnsmasq.conf.**

Step 8: Edit the configuration file as shown in the Figure 47.

Step 9: Configure the **hosts** file with the command **sudo nano /etc/hosts**

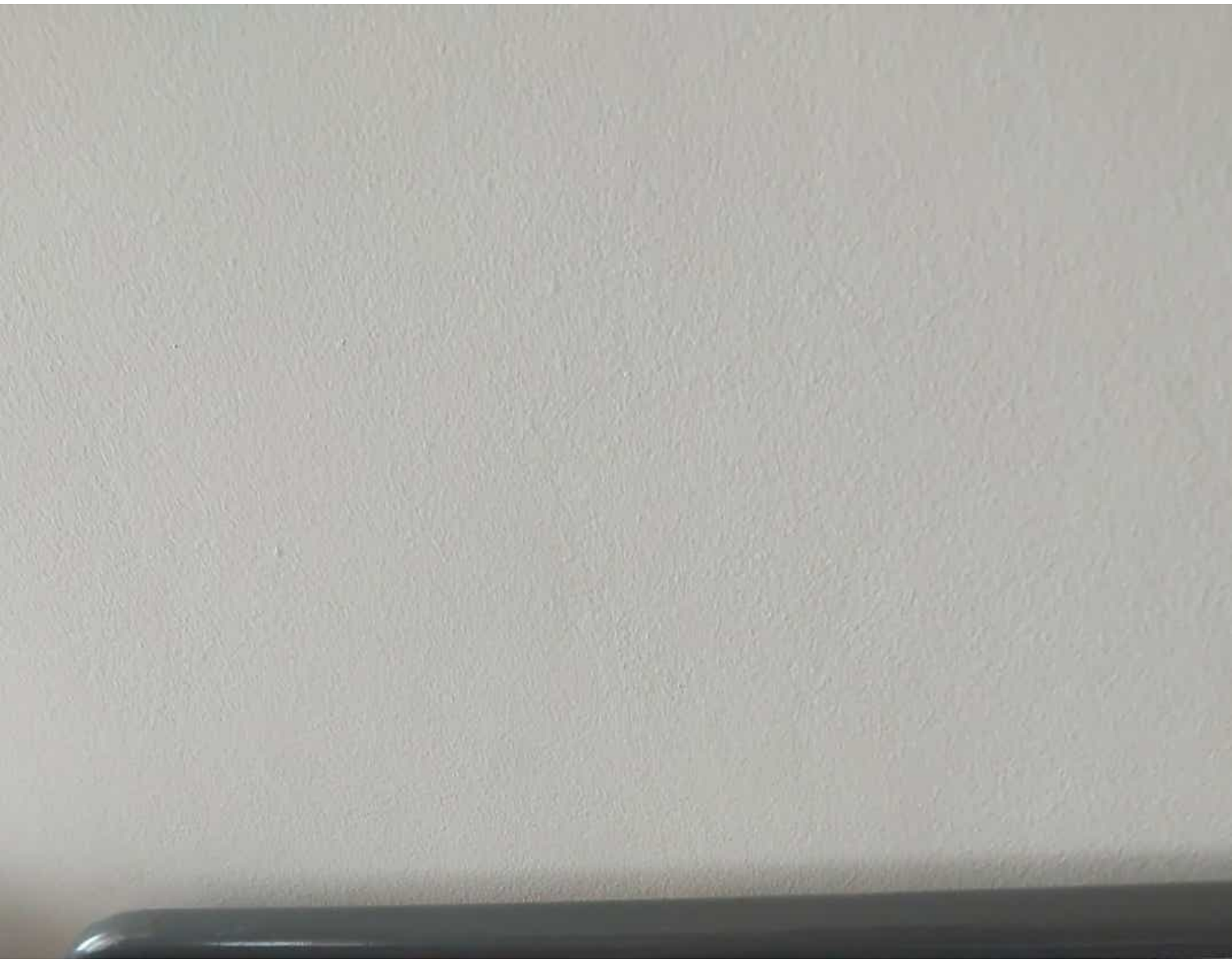
Step 10: Edit the configuration file as shown in the Figure 47.

Step 11: Restart the services of **dnsmasq** with the command **sudo systemctl restart dnsmasq**

Step 12: Use the command **dig a deepikak.in @localhost** to fetch the IP specified for the mentioned domain in the **hosts** file.

Expected Output

- The students will be able to configure DNS Server
- The students will be able to rename a DNS Server and obtain the IP address



Program - 7

Build a Firewall to Restrict Network Access using Firewall or Build a firewall with SNAT or DNAT

Description

Create a private LAN with minimum two machines. Firewall policy is to define which traffic should be permitted into the organization's networks and hence be very careful while implementing it.

Firewall

Firewall is one of the method to provide network security which can be implemented either through software or hardware. The simple way is to provide the security at network layer. System Administrators provide security to private network by blocking (deny) or allowing (permit) the packets to cross the firewall based on the user defined rules.

iptables

iptables is an Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. NAT into two different types: Source NAT (SNAT) and Destination NAT (DNAT).

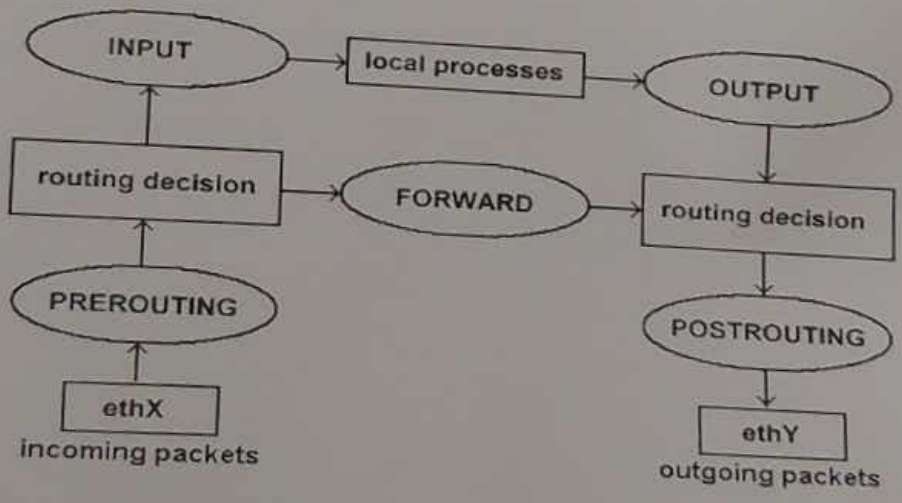


Figure 48: Representation of **iptables** with routing

Required Components

- Computers or Laptops - 02
- LAN Cables - 02

Network Diagram & Procedure

1. Connect the Network

STEP 1

Computer 1
Username: ubuntu1
IP: 180.0.0.15
Netmask: 255.255.0.0

IP: 180.0.0.1

Computer 2
Username: ubuntu2
IP: 180.0.0.20
Netmask: 255.255.0.0

2. System Preparation

STEP 2

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the iptables Software using the command

sudo apt-get install iptables

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the iptables Software using the command

sudo apt-get install iptables

STEP 3

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

ACCEPT traffic from a specific IP address

sudo iptables -A INPUT -s 180.0.0.15 -j ACCEPT

DROP traffic from an IP address with

sudo iptables -A INPUT -s 180.0.0.15 -j DROP

REJECT traffic from a range of IP addresses with

sudo iptables -A INPUT -m iprange --src-range 180.0.0.10-180.0.0.25 -j REJECT

sudo iptables -F

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Validate the Firewall with

ping 180.0.0.15

Figure 49: Network Diagram & Procedure

Step 1: Connect the Server System to Internet using Ethernet or Wi-Fi connection and update the system.

Step 2: Install iptables on the Server using the command - **sudo apt-get install iptables**.

Step 3: Disconnect the Server Machine from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Server using the command - **sudo ifconfig**.

Step 4: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services.

Step 5: Set the IP address using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**.

```
deepika@debian:~$ sudo apt-get install iptables                    Step 2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.7-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
deepika@debian:~$ sudo systemctl stop NetworkManager
deepika@debian:~$ sudo ifconfig enp0s25 180.0.0.25 netmask 255.255.0.0 up
deepika@debian:~$ sudo ifconfig enp0s25
enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 180.0.0.25  netmask 255.255.0.0  broadcast 180.0.255.255
        inet6 fe80::3e97:eff:fee4:6ecf  prefixlen 64  scopeid 0x20<link>
        ether 3c:97:0e:e4:6e:cf  txqueuelen 1000  (Ethernet)
        RX packets 7  bytes 502 (502.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 5600 (5.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf2500000-f2520000

deepika@debian:~$ ping 180.0.0.15
PING 180.0.0.15 (180.0.0.15) 56(84) bytes of data.
64 bytes from 180.0.0.15: icmp_seq=1 ttl=64 time=7.93 ms
64 bytes from 180.0.0.15: icmp_seq=2 ttl=64 time=3.69 ms
^C
--- 180.0.0.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 3.693/5.811/7.930/2.118 ms
deepika@debian:~$
```

Figure 50: Installation of **iptables**

Source NAT

Source NAT; change the source address of connections to something different. This is done in the POSTROUTING chain, just before it is finally sent out; this is an important detail, since it means that anything else on the GNU Linux box itself (routing, packet filtering) will see the packet unchanged. It also means that the '-o' (outgoing interface) option can be used.

Step 6: Use the following command to ACCEPT traffic from a specific IP address. with the command **sudo iptables -A INPUT -s 180.0.0.15 -j ACCEPT**

Step 7: DROP traffic from an IP address with **sudo iptables -A INPUT -s 180.0.0.15 -j DROP**

Step 8: REJECT traffic from a range of IP addresses with **sudo iptables -A INPUT -m iprange –src-range 180.0.0.10-180.0.0.25 -j REJECT**

Step 9: Use the -F option to clear all iptables firewall rules. A more precise method is to delete the line number of a rule. **sudo iptables -L –line-numbers**

```
deepika@debian:~$ sudo iptables -A INPUT -s 180.0.0.15 -j DROP
deepika@debian:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  180.0.0.15          anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
deepika@debian:~$ sudo iptables -F
deepika@debian:~$ sudo iptables -A INPUT -m iprange --src-range 180.0.0.10-180.0.0.20 -j REJECT
deepika@debian:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            source IP range 180.0.0.10-180.0.0.20 reject-wit
h icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
deepika@debian:~$
```

Figure 51: Execution of Source NAT

Destination NAT

Destination NAT; is done in the PREROUTING chain, just as the packet comes in; this means that anything else on the Linux box itself (routing, packet filtering) will see the packet going to its 'real' destination. It also means that the '-i' (incoming interface) option can be used.

Destination NAT is specified using '-j DNAT', and the '–to-destination' option specifies an IP address, a range of IP addresses, and an optional port or range of ports (for UDP and TCP protocols only).

Step 9: Change destination addresses to 120.0.0.40 using **sudo iptables -t nat -A PREROUTING -i eth0 -j DNAT –to 120.0.0.40**

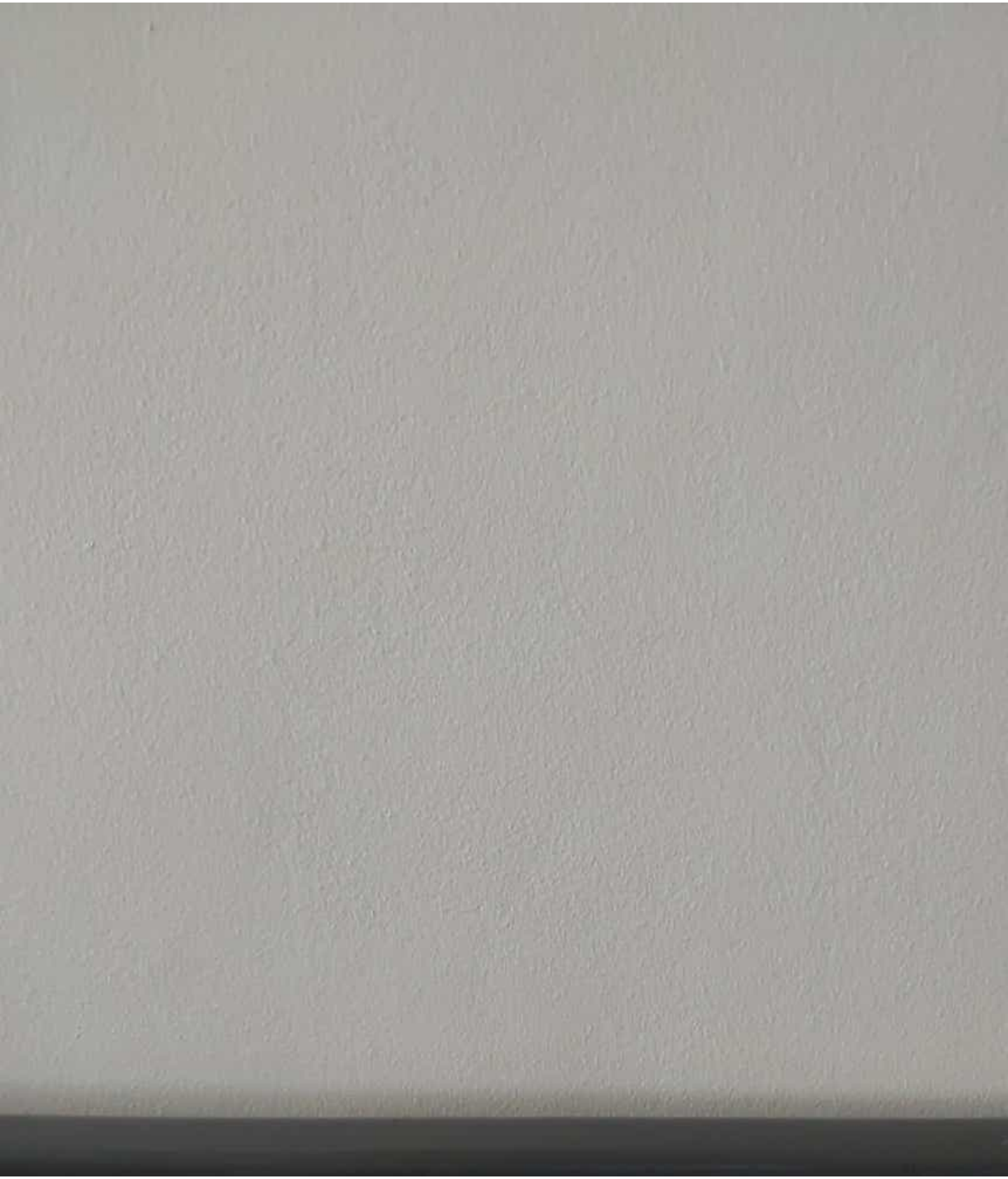
Step 10: Change destination addresses to 120.0.0.40, 120.0.0.50, 120.0.0.60 using **sudo iptables -t nat -A PREROUTING -i eth0 -j DNAT –to 120.0.0.40-120.0.0.60**

Step 11: Change source addresses to 120.0.0.10, ports 1-1023 using the command **sudo iptables -t nat -A PREROUTING -p tcp –dport 80 -i eth0 -j DNAT –to 120.0.0.40:8080**

Step 12: Execute PING command after every step and flush the iptables using **-F** after every Step.

Expected Output

- Students will be able to build firewalls, planning firewall design and implementation to meet security needs, configuring firewalls in alignment with a firewall policy



Program - 8

Demonstrate basic trouble shooting using ping, traceroute, ifconfig, nslookup, netstat and route

Description

GNU Linux Operating system consists of various built-in, command-line networking utilities that are used for network troubleshooting. Some basic trouble shooting networking commands which are most essentials for every network administrator.

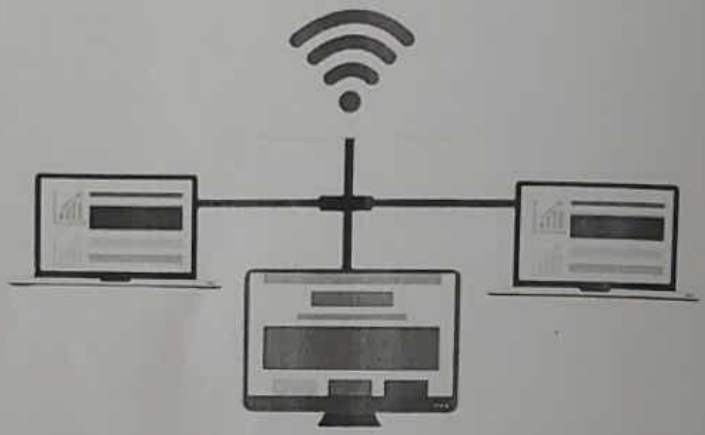


Figure 52: Representation of Network Connectivity to Internet

traceroute

A traceroute works by sending Internet Control Message Protocol (ICMP) packets, and every router involved in transferring the data gets these packets. The ICMP packets provide information about whether the routers used in the transmission are able to effectively transfer the data.

ping

Ping is used to testing a network host capacity to interact with another host. The **ping** command followed by the target host's name or IP address. The ping utilities seem to be the most common network tool. This is performed by using the Internet Control Message Protocol, which allows the echo packet to be sent to the destination host and a listening mechanism. If the destination host reply to the requesting host, that means the host is reachable. This utility usually gives a basic image of where there may be a specific networking issue.

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date 24-4-23 Name Priyanka S P

Dept./Lab CN Class I, B Expt./No. 8

Title Demonstrate basic route shooting using ping, traceroute, ifconfig, nslookup, netstat & route

* ping <ip-address>

→ To check the connectivity of system in the network.

$ ping <ip-address>

* traceroute

Displays the count of number of hops used to transmit the packets

$ traceroute <domain-name>

* ifconfig

Displays network configurations of system

$ ifconfig

Signature of Teacher incharge

nslookup :-

Resolves the IP address fo
given domain name

$ nslookup <ip-address>

* netstat :-

Displays the network statistic

$ netstat

* route

Displays the routing table.

$ route

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date _____ Name _____

Dept./Lab _____ Class _____ Expt./No. _____

Title _____

8 Exp

PING

1) Ping 127.0.0.1

2) Ping -c 5 127.0.0.1] only 5 packets are sent.

3) Ping -4 127.0.0.1] use IPv4 address only.

4) Ping -D 127.0.0.1]

print timestamp (unix time + microseconds) before each line.

O/P:- [1682321172.177695] 64 bytes from

127.0.0.1 icmp_seq=1 ttl=64 time=0.055ms

5) Ping -i 1 127.0.0.1

waits for 1 second between sending each packet

O/p:- 64 bytes from 127.0.0.1 icmp_seq=2 ttl=64

time = 0.062ms.

B) ifconfig

Signature of
Teacher incharge

Address : 2404 : 6800 : 4007 : 820 : : 20

E) $route

$ route -A inet
use the specified address family

Destination gateway genmask Flags metric Ref
default gateway 0.0.0.0 UG 100 0
use Iface
0 enp1s0.

$ route -n
Show numerical address instead of determining
symbolic hosts name

Destination gateway genmask Flags metric Ref
0.0.0.0 172-16.35-254 0.0.0.0 UG 100 0
use Iface
0 enp1s0

$ route -e
use netstat(8) - format for displaying.

R.V. COLLEGE OF ENGINEERING®

OBSERVATION / DATA SHEET

Date _____ Name _____

Dept./Lab _____ Class _____ Expt./No. _____

Title _____

F) traceroute :-

$ traceroute 172.16.34.30

172.16.34.30 0.008 m/s 0.006 ms 0.005~

$ traceroute -f 12 172.16.34.30.

To set initial hop distance, ie: time to live

172.16.34.30 0.008 ms 0.006 ms 0.005~

$ traceroute -q 4 172.16.34.30.

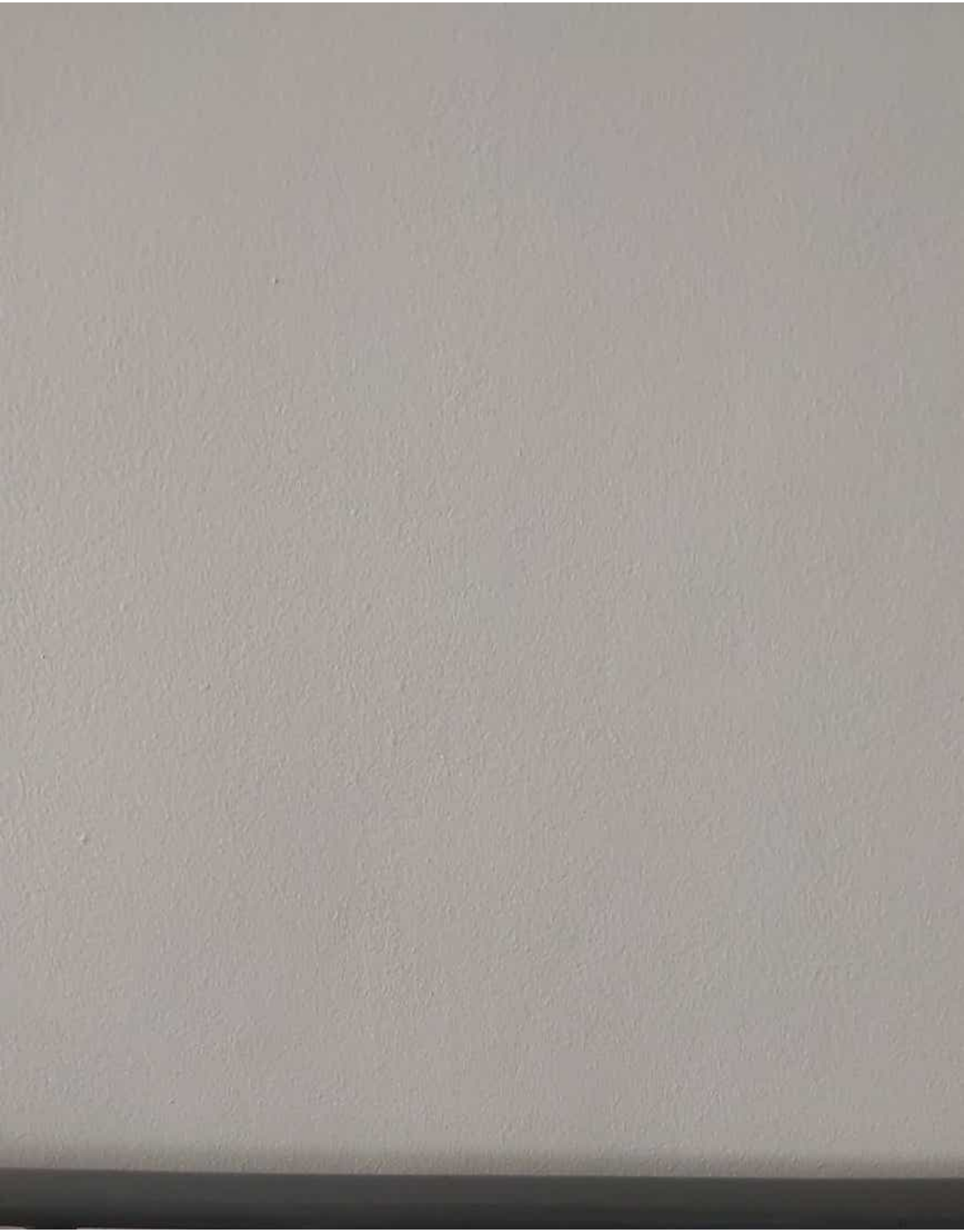
To send n probe packets per hop.

1 172.16.34.30 0.002 m/s 0.016 ms 0.002

0.001 ms.

— x —

Signature of
Teacher incharge



Program - 9

Demonstrate multiple client server communication on different ports using netcat

Description

Netcat (nc) command is a command-line utility for reading and writing data between two computer networks. The communication happens using either TCP or UDP. The command differs depending on the system (netcat, nc, ncat, and others). Netcat is a crucial tool to master for network and system administrators due to the rich connection troubleshooting features and scripting usability. NetCat is represented in Figure 59.

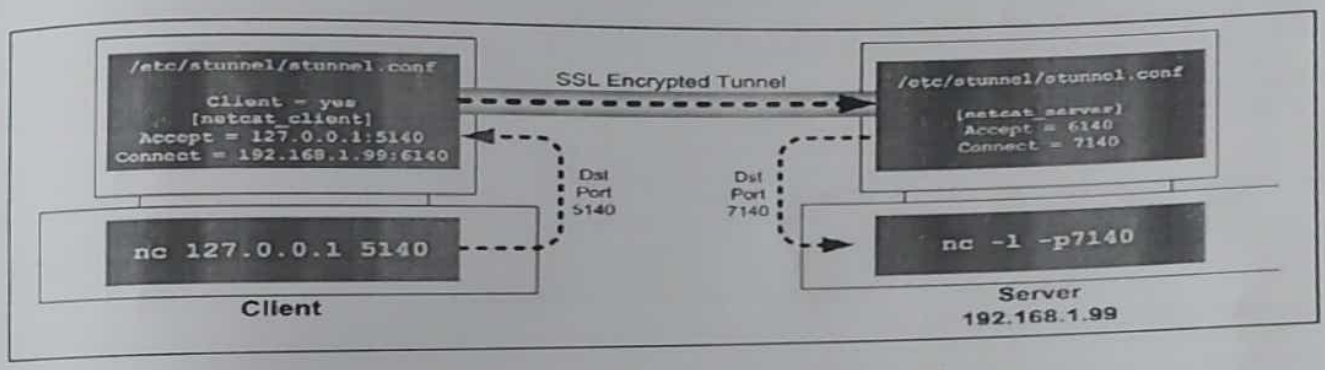


Figure 59: Representation of **NetCat (nc)**

Option	Type	Description
-4	Protocol	Use IPv4 only.
-6	Protocol	Use IPv6 only.
-u –udp	Protocol	Use UDP connection.
-p <port> –source-port <port>	Connect mode	Binds the Netcat source port to <port>.
-s <host> –source <host>	Connect mode	Binds the Netcat host to <host>.
-l –listen	Listen mode	Listens for connections instead of using connect mode.
-z	Output	Report connection status without establishing a connection.

Required Components

- Computers or Laptops - 02
- Manageable Router / Switch - 01
- LAN Cables - 02

Network Diagram

Step 1: Make a network diagram as shown in 43 with necessary information such as Username, Ip and Netmask for all the machines.

Step 2: Connect the systems to Internet using Ethernet or Wi-Fi connection and update the system

Step 3: Install netcat on the machines using the command - **sudo apt-get install netcat**

```
deepika@debian:~$ sudo apt-get install netcat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
netcat is already the newest version (1.10-46).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
deepika@debian:~$ sudo systemctl stop NetworkManager
deepika@debian:~$ sudo ifconfig enp0s25 190.0.0.5 netmask 255.255.255.0 up
[sudo] password for deepika:
deepika@debian:~$ sudo ifconfig enp0s25
enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 190.0.0.5  netmask 255.255.255.0  broadcast 190.0.0.255
        ether 3c:97:0e:e4:6e:cf  txqueuelen 1000  (Ethernet)
        RX packets 97  bytes 8628 (8.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 97  bytes 14276 (13.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf2500000-f2520000

deepika@debian:~$ nc -lv 1234
Listening on 0.0.0.0 1234
Connection received on 190.0.0.10 59410
Hello from Server
Hi from Client
deepika@debian:~$
```

Figure 61: Procedure of **NetCat (nc) Client-Server Communication**

Step 4: Disconnect the Server Machine from Internet. Open the Terminal (Ctrl + Alt + T) and observe the ethernet interface on the Server using the command - **sudo ifconfig**

Step 5: Type the command - **sudo systemctl stop NetworkManager** to stop the networking services

Step 6: Set the IP address using - **sudo ifconfig <ethernet_interface_id> <ip> netmask <netmask> up**

Step 7: Check the IP Address using the command - **sudo ifconfig**

Step 8: Client/Server Connection is between two devices. One device acts as a server (listens) while the other acts as a client (connects). Run the command - **nc -lv 1234** on the Server. The -l option activates listen mode, making the machine the server. The output shows the device listening for connections due to the -v option.

1. Connect the Network

STEP 1

Computer 1
Username: ubuntu1
IP: 190.0.0.5
Netmask: 255.255.0.0

IP: 190.0.0.1

Computer 2
Username: ubuntu2
IP: 190.0.0.10
Netmask: 255.255.0.0

2. System Preparation

STEP 2

Connect the System to Internet with Wired or Wireless Connectivity	Connect the System to Internet with Wired or Wireless Connectivity
Update the System using the command **sudo apt-get update**	Update the System using the command **sudo apt-get update**
Install the netcat Software using the command **sudo apt-get install netcat**	Install the netcat Software using the command **sudo apt-get install netcat**

3. Setting up the IP

STEP 3

Disconnect the Internet Connectivity	Disconnect the Internet Connectivity
Open the Terminal (Ctrl + Alt + T) Type the command - **sudo ifconfig**	Open the Terminal (Ctrl + Alt + T) Type the command - **sudo ifconfig**
Observe the ethernet interface. Find the ID of the ethernet interface card and stop the networking services **sudo systemctl stop NetworkManager**	Observe the ethernet interface. Find the ID of the ethernet interface card and stop the networking services **sudo systemctl stop NetworkManager**
Run the nc command with the IP address of the Server **nc -v 190.0.0.10 1234**	Run the nc command with port number **nc -lv 1234**
Initiate Client communication using TCP with netcat via Terminal	Initiate Server communication using TCP with netcat via Terminal

-v = remove

Figure 60: Network Diagram & Procedure

Step 9: Run the nc command on Client with the IP address of Server and the port - `nc -v 190.0.0.10 1234`

Step 10: Initiate Client Server communication using TCP with netcat via Terminal. Multiple Clients can also establish communication with the Server

Expected Output

- The students will be able to implement multiple Client Server communication using netcat and understand the working of modes - connect and listen modes
- The students will be able to understand the working of tcp with netcat command-line utility

Sl. No	Rubrics for Practice Session		Total Marks	Obtained Marks
1	**Conduction & Execution**	Understanding of Concepts & logic	02	2
2		Approach Towards the Problem	02	2
3		Conduction and Execution: Input & Output for allpossible cases	03	2.5
4	**Viva Voce**	Program Analysis & Applications	02	2
5		Communication & Confidence Level	01	1
Total			10	9.5

dbd
15/10/23

Program - 10

Demonstrate Proxy - Server setup for a web server and SSH port forwarding

Description

A Proxy Server verifies and forwards incoming client requests to other servers for further communication. A proxy server is located between a client and a server where it acts as an intermediary between the two, such as a Web browser and a web server. The proxy server's most important role is providing security.The basic representation of Proxy-Server is shown in Figure 62.

Some people use proxies for personal purposes, such as hiding their location while watching movies online, for example. For a company, however, they can be used to accomplish several key tasks such as:

1. Improve security
2. Secure employees' internet activity from people trying to snoop on them
3. Balance internet traffic to prevent crashes
4. Control the websites employees and staff access in the office
5. Save bandwidth by caching files or compressing incoming traffic

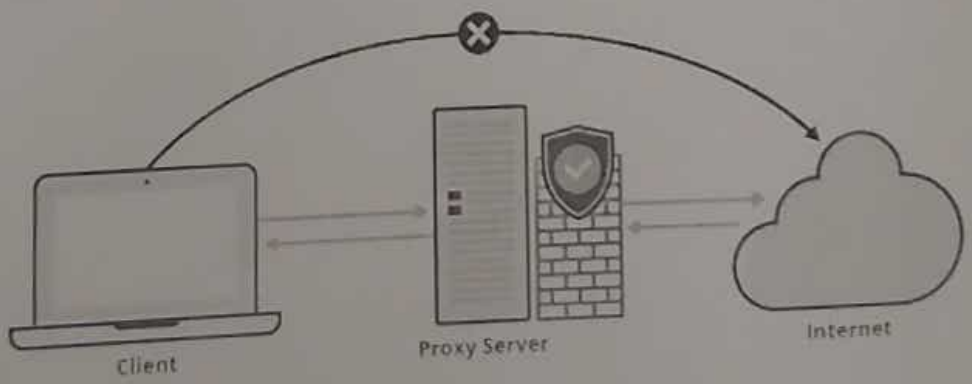


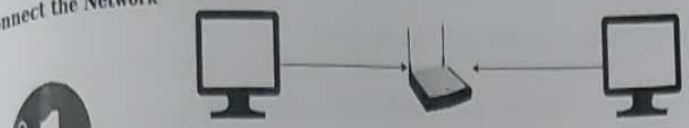
Figure 62: Proxy Server

Required Components

- Computers or Laptops - 02
- Manageable Router / Switch - 01
- LAN Cables - 02

Network Diagram

1. Connect the Network



Computer 1
Username: ubuntu1
IP: 192.168.0.100
Netmask: 255.255.255.0

IP: 192.168.0.1

Computer 2
Username: ubuntu2
IP: 192.168.0.200
Netmask: 255.255.255.0

2. System Preparation

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the Squid and Lynx Softwares using the command

sudo apt-get install squid
sudo apt-get install lynx

Connect the System to Internet with Wired or Wireless Connectivity

Update the System using the command

sudo apt-get update

Install the Squid and Lynx Software using the command

sudo apt-get install squid
sudo apt-get install lynx

3. Setting up the IP

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Create a directory and a simple index.html

mkdir /tmp/website
cd /tmp/website/
echo "Hi there" > index.html

Make sure python3 is installed and then run:

cd /tmp/website/
python3 -m http.server 8000 --bind 127.0.0.1

Now client is running a webserver on 127.0.0.1:8000 that is not accessible from outside the computer.

Disconnect the Internet Connectivity

Open the Terminal (Ctrl + Alt + T)
Type the command -

sudo ifconfig

Observe the ethernet interface.
Find the ID of the ethernet interface card and stop the networking services

sudo systemctl stop NetworkManager

Initiate a SSH tunnel (port forward) to access the webserver

ssh ubuntu1@192.168.0.100 -v -L 127.0.0.1:6000:127.0.0.1:8000

Access the following url using a web browser

http://127.0.0.1:6000/

See "Hi there"

Figure 63: Network Diagram & Procedure

Step 1: Make a network diagram as shown in 43 with necessary information such as Username, IP and Netmask for all the machines.

Step 2: Connect the systems to Internet using Ethernet or Wi-Fi connection and update the system

Step 3: Install Squid with **sudo apt-get install squid**

Step 4: Install Lynx with **sudo apt-get install lynx**

Step 5: Open gnu.org from the terminal **lynx http://gnu.org**

```
deepika@debian:~$ sudo apt-get install squid
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
squid is already the newest version (4.13-10+deb11u2).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
deepika@debian:~$ sudo apt-get install lynx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lynx is already the newest version (2.9.0dev.6-3-deb11u1).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.

deepika@debian:~$ lynx https://gnu.org
```

Figure 64: Installation of Lynx & Squid, Opening a Webpage from Terminal

```
                                                    The GNU Operating System and the Free Software Movement (p1 of 9)

Skip to main text

The future of sharing is up to you! Join the FSF by Dec 31 to defend your freedom to share.

READ MORE

 [A GNU head]  GNU Operating System

 [Search www.gnu.org]
LANGUAGES
Site navigation Skip
  * ABOUT GNU
  * PHILOSOPHY
  * LICENSES
  * EDUCATION
  * SOFTWARE
  * DISTROS
  * DOCS
  * MALWARE
  * HELP GNU
-- press space for next page --
```

Figure 65: Output of **lynx http://gnu.org**

Step 6: On the Server machine, create a directory and a simple index.html file
mkdir /tmp/website
cd /tmp/website/
echo "Hi there" > index.html

Step 7: Make sure python3 is installed and then run **cd /tmp/website/**
python3 -m http.server 8000 –bind 127.0.0.1

Step 8: Now the Server machine, is running a web server on **127.0.0.1:8000** that is not accessible from outside the computer.

Step 9: The Client machine can initiate a SSH tunnel (port forward) to access the webserver by SSH to the Server **ssh user@192.168.0.15 -v -L 127.0.0.1:6000:127.0.0.1:8000**. Access the following URL using the command - **lynx http://127.0.0.1:6000/**

```
deepika@debian:~$ mkdir /tmp/website                                    Step 6
deepika@debian:~$ cd /tmp/website/
deepika@debian:/tmp/website$ echo "Hi there" > index.html
deepika@debian:/tmp/website$ cat index.html
Hi there                                                                Step 7
deepika@debian:/tmp/website$ python3 -m http.server 8000 --bind 127.0.0.1
Serving HTTP on 127.0.0.1 port 8000 (http://127.0.0.1:8000/) ...
```

Figure 66: Using Python to run the Server

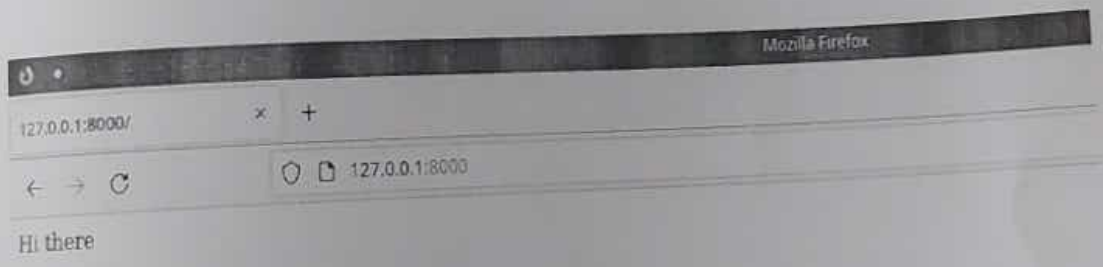


Figure 67: Output of **Proxy-Server**

Expected Output

- The students will be able to setup the Proxy - Server using Squid and Lynx
- The students will be able to demonstrate access to location-specific content and provide enhanced security