Department of Computer Engineering

TE: SEM V Subject: CN

Experiment: 03

Aim:

a. Using netstat and route commands of Linux, do the following:

- View current routing table
- Add and delete routes
- Change default gateway

b. Perform packet filtering by enabling IP forwarding using IPtables in Linux.

Description:

Routing is the transfer of an IP packet from one point to another across the network. When you send someone an email, you're actually transmitting a series of IP packets or datagrams from your system to the other person's computer. The packets sent from your computer pass through several gateways or routers to get to the destination computer system. The same is true for all Internet protocols such as HTTP, IRC, FTP, etc.

In all Linux and UNIX systems, the information about how the IP packets should be routed is stored in a kernel structure. These structures are called routing tables. If you want your system to communicate with other computers, you may want to configure these routing tables. First, it is important to know how to view these routing tables on your Linux system.

To view the routing tables in Ubuntu using the following three common commands:

- The netstat command
- The route commands
- The ip route command

Method 1: Through the netstat command

The netstat command has always been a widely used method of printing routing table information in Linux. However, it is officially replaced by the ip route command. We are including it anyway as it is still an approach to retrieve the required information.

Here is how you can use this command:

```
$ netstat -rn
```

-r This flag is used to display the Kernel routing tables

-n This flag is used to display the numerical addresses

```
sofiya@LAPTOP-NT7PQD1K:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U         0 0          0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
0.0.0.0         192.168.192.193 255.255.255.255 U         0 0          0 wifi0
192.168.0.0     0.0.0.0         255.255.0.0     U         0 0          0 wifi0
192.168.2.154   0.0.0.0         255.255.255.255 U         0 0          0 wifi0
192.168.255.255 0.0.0.0         255.255.255.255 U         0 0          0 wifi0
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 wifi0
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 wifi0
```

This is what the output indicates:

Destination	This column indicates the destination network.
Gateway	This column indicates the defined gateway for the network. If you see an * in this column, it means that no forwarding gateway is needed for the specified network.
Genmask	This column indicates the netmask of the network.
Flags	The U output in this columns means that the route is up. The G output indicates that specified gateway should be used for this route. D stands for dynamically installed, M stands for modified, and R means reinstated. The H flag indicates that the destination is a fully qualified host address, rather than a network.
MSS	This column indicates the default Maximum Segment Size(MSS) for TCP connections for this route.
Window	This column indicates the default window size for TCP connections over this route.
Irtt	This column indicates the Initial Round Trip Time for this route.
Iface	The Iface column shows the network interface. If you had more than one interface, you would see *lo* (for loopback), *eth0* (first Ethernet device), and *eth1* (for the second Ethernet device), and so on for the number of interfaces, you have installed.
matric	The metric field has a number of different meanings: The Metric field indicates the cost of a route. If multiple routes exist to a given destination network ID, the metric is used to decide which route is to be taken. The route with the lowest metric is the preferred route.

Method 2: Through the route command

The route command also falls under the category of once widely used but now obsolete command to view routing tables. The manual page of this command also mentions that the command is now replaced by the ip route command.

Through this command, you can view exactly the same information that you could, through the netstat command. Here is how you can use it:

```
$ route -n
```

-n This flag is used to display the numerical addresses only

```
sofiya@LAPTOP-NT7PQDIR:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     256    0        0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U     256    0        0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
0.0.0.0         192.168.192.193 255.255.255.255 U     0      0        0 wifi0
192.168.0.0     0.0.0.0         255.255.0.0     U     256    0        0 wifi0
192.168.2.154   0.0.0.0         255.255.255.255 U     256    0        0 wifi0
192.168.255.255 0.0.0.0         255.255.255.255 U     256    0        0 wifi0
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 wifi0
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 wifi0
```

Method 3: Through the ip route command

Last but not least, here is the most recommended way of printing routing table information in Linux. Here is how to use this command:

```
$ ip route
```

```
sofiya@LAPTOP-NT7PQD1K:~$ ip route
none 224.0.0.0/4 dev eth0 proto unspec metric 256
none 255.255.255.255 dev eth0 proto unspec metric 256
none 127.0.0.0/8 dev lo proto unspec metric 256
none 127.0.0.1 dev lo proto unspec metric 256
none 127.255.255.255 dev lo proto unspec metric 256
none 224.0.0.0/4 dev lo proto unspec metric 256
none 255.255.255.255 dev lo proto unspec metric 256
none default via 192.168.192.193 dev wifi0 proto unspec metric 0
none 192.168.0.0/16 dev wifi0 proto unspec metric 256
none 192.168.2.154 dev wifi0 proto unspec metric 256
none 192.168.255.255 dev wifi0 proto unspec metric 256
none 224.0.0.0/4 dev wifi0 proto unspec metric 256
none 255.255.255.255 dev wifi0 proto unspec metric 256
none 224.0.0.0/4 dev wifi1 proto unspec metric 256
none 255.255.255.255 dev wifi1 proto unspec metric 256
none 224.0.0.0/4 dev wifi2 proto unspec metric 256
none 255.255.255.255 dev wifi2 proto unspec metric 256
```

Though this information is not much reader-friendly as that of the previously mentioned commands, it is still enough for you to configure the router.

To add route in table—

First we will check current routing table

```
apsit@apsit-HP-280-G2-SFF:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp2s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.192.193 0.0.0.0         UG    100    0        0 enp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
192.168.0.0     0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
apsit@apsit-HP-280-G2-SFF:~$
```

Adding a static route using IP command

Suppose you want to take a backup of a Linux machine and push the backup file to another backup server in the subnet **10.0.2.0/24**. However, for one reason or the other, you cannot reach the backup server via the default gateway. In this case, you

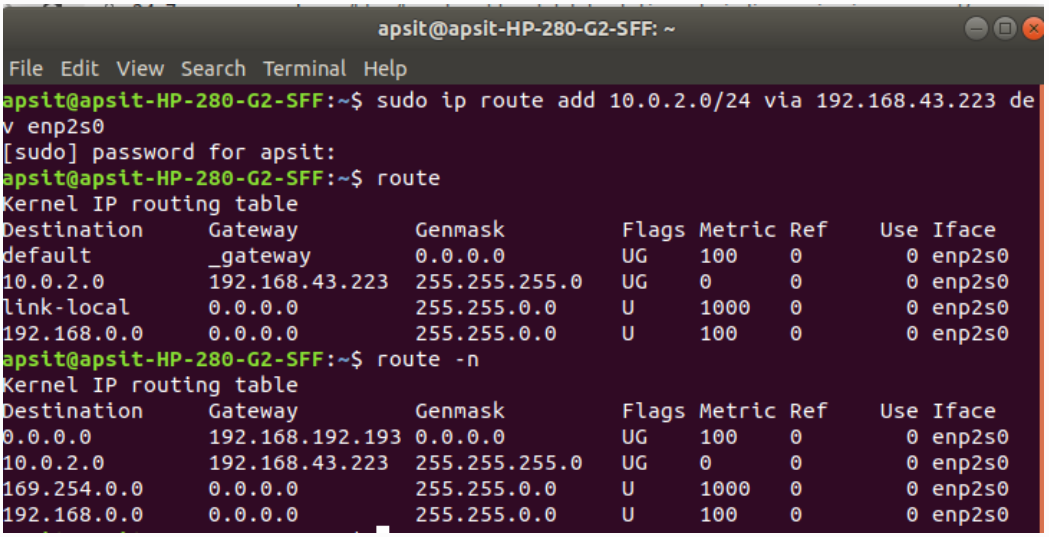
will have to create a new route for the backup server subnet via another IP, say **192.168.43.223** via the interface **enp0s3**.

The command for this will be

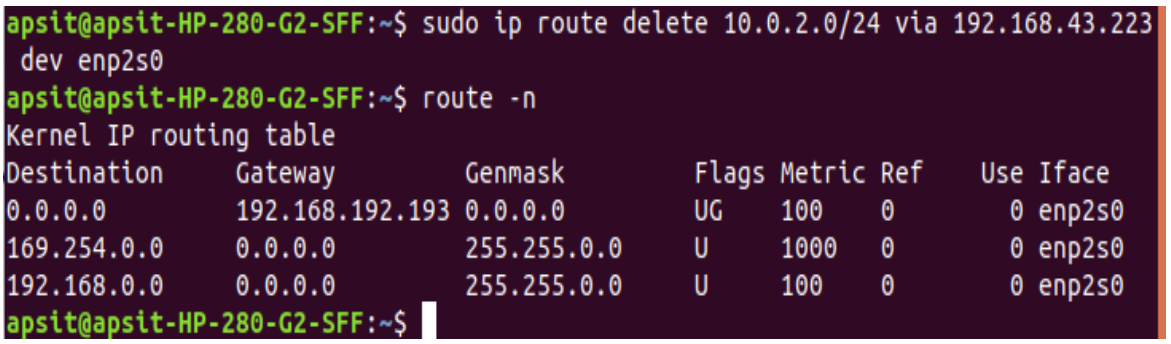
$ sudo ip route add 10.0.2.0 via 192.168.43.223 dev enp2s0

Where:

- 10.0.2.0 -> is the network you want to connect to
- /24 -> is the subnet mask
- 192.168.43.223 -> is the IP through which we will reach the server
- enp2s0 -> is the network interface



Deleting a static route using IP command



To set default gateway-

ip command to set a default router to 192.168.1.254

Login as the root and type:

ip route add default via 192.168.1.254

OR

$ sudo ip route add default via 192.168.1.254

route command to set a default router to 192.168.1.254

Login as the root and type:

route add default gw 192.168.1.254

OR

$ sudo route add default gw 192.168.1.254

To perform packet filtering by enabling IP forwarding using IPtables in Linux

A user-space Unix utility that gives system administrators the ability to configure IP packet filtering rules implemented by the Kernel's net filter module.Iptables act as a firewall using packet filtering rules based on various criteria such as IP address, port, and protocols.
Iptables come pre-installed on Ubuntu and most Debian based distributions. Ubuntu also packages GUFW firewall, a graphical alternative you can use to work with iptables.

The Filter Tables

The filter table is a default table that contains chains used for network packet filtering. Some of the default chains in this table include:

Input	Iptables use this chain for any incoming packets to the system, i.e., packets going to local network sockets.
Output	Iptables use the output chain for locally generated packets, i.e., packets going out of the system.
Forward	This chain is what the Iptables use for packets routed or forwarded via the system.

1.To check current iptable rule-

iptables -L

2.To drop forward chain-(It drops packets for router)

sudo iptables -P FORWARD DROP

3.To drop packets incoming from specific ip address-

iptables -A INPUT -s 192.168.0.23 -j DROP

4.Consider the command below:To drop packets incoming from specific ip address-

```
sudo iptables -I INPUT -s 192.168.0.24 -j DROP
```

The command above tells the iptables to create a rule in the chain. The rule drops all the packets from the IP address 192.168.0.24.

Let us examine the command, line by line, to understand it better.

- The first command iptables calls the iptables command-line utility.
- Next is -I argument used for insertion. The insertion argument adds a rule at the beginning of the iptables chain and thus gets assigned a higher priority. To add a

rule at a specific number in the chain, use the -I argument followed by the number where the rule should get assigned.

- The -s argument helps specify the source. Hence, we use the -s argument followed by the IP address.
- The -j parameter with iptables specifies the jump to a specific target. This option sets the action the Iptables shall perform once there's a matching packet. Iptables offers four main targets by default, these include: ACCEPT, DROP, LOG(Use –log-level followed by a number to define the level of LOG provided by Iptables), and REJECT.

5.To drop packets from particular network-(For SMTP Port-25)

iptables -A INPUT -s 192.168.0.0/24 -p tcp - -destination-port 25 -j DROP

6.To accept particular packets from specific network

iptables -A INPUT -s 192.168.0.66 -j ACCEPT