PES UNIVERSITY
EC CAMPUS, BANGALORE

Name – B. Pravena

SRN – PES2UG19CS076

WEEK – 1

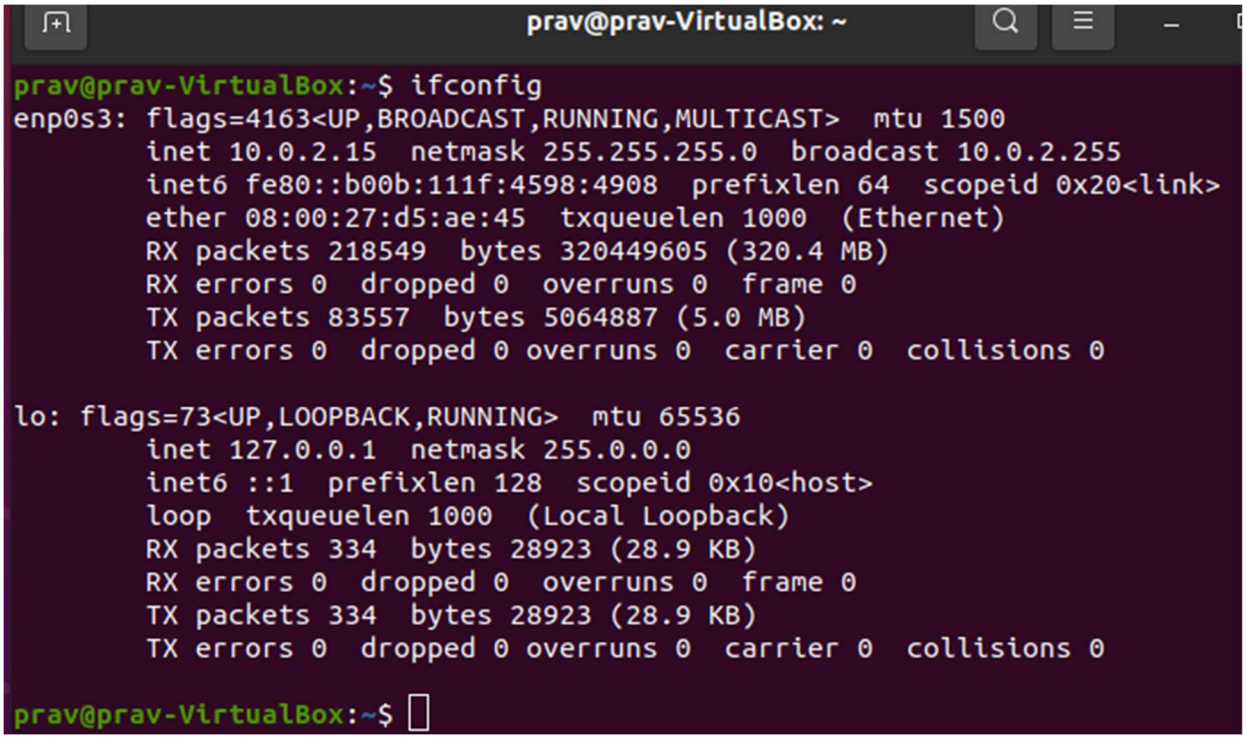
SUBJECT - Computer Networks Laboratory

Objective -: To study and understand the basic networking tools - wireshark, Tcpdump, Ping, Traceroute and Netcat.

Task 1: Linux Interface Configuration (ifconfig / IP command)

Step 1: To display status of all active network interfaces.

ifconfig (or) **ip addr show**



Analyze and fill the table-:

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address
enp0s3	IPV4 - 10.0.2.15 IPV6 – fe80::b00b:111f:4598:4908	ether 08:00:27:d5:ae:45
lo	IPV4 - 127.0.0.1 IPV6 - ::1	00:00:00:00:00:00

Step 2: To assign an IP address to an interface, use the following command.

sudo ifconfig lo 10.0.2.10 netmask 255.255.255.0

```
prav@prav-VirtualBox:~$ sudo ifconfig lo 10.0.2.10 netmask 255.255.255.0
prav@prav-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b00b:111f:4598:4908  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d5:ae:45  txqueuelen 1000  (Ethernet)
        RX packets 218560  bytes 320450767 (320.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83574  bytes 5066410 (5.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 10.0.2.10  netmask 255.255.255.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 351  bytes 31153 (31.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 351  bytes 31153 (31.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

prav@prav-VirtualBox:~$
```

The IP address of “lo” has changed from previous.

Step 3: Activate/deactivate a network interface, type.

sudo ifconfig lo down

```
prav@prav-VirtualBox:~$ sudo ifconfig lo down
prav@prav-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b00b:111f:4598:4908  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d5:ae:45  txqueuelen 1000  (Ethernet)
        RX packets 218576  bytes 320451847 (320.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83605  bytes 5069084 (5.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

prav@prav-VirtualBox:~$
```

sudo ifconfig lo up

```
prav@prav-VirtualBox:~$ sudo ifconfig lo up
prav@prav-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b00b:111f:4598:4908  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d5:ae:45  txqueuelen 1000  (Ethernet)
        RX packets 218590  bytes 320453041 (320.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83625  bytes 5070811 (5.0 MB)
        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 367  bytes 33323 (33.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 367  bytes 33323 (33.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

prav@prav-VirtualBox:~$
```

Step 4: To show the current neighbor table in kernel, type

ip neigh

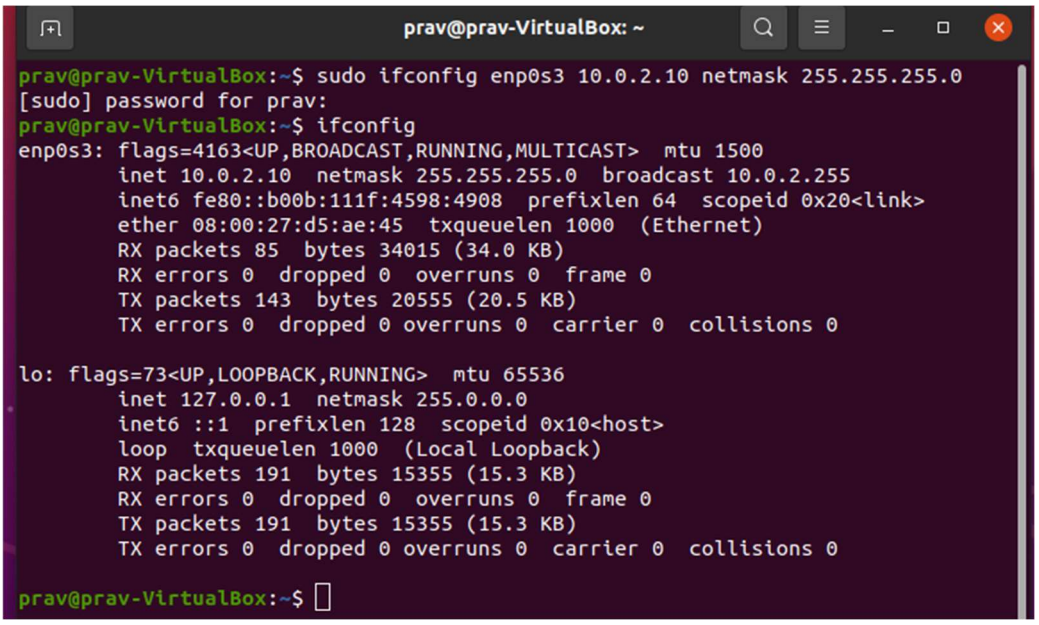
When the wired connection is ON and OFF respectively.

```
prav@prav-VirtualBox:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 STALE
prav@prav-VirtualBox:~$ ip neigh
prav@prav-VirtualBox:~$
```

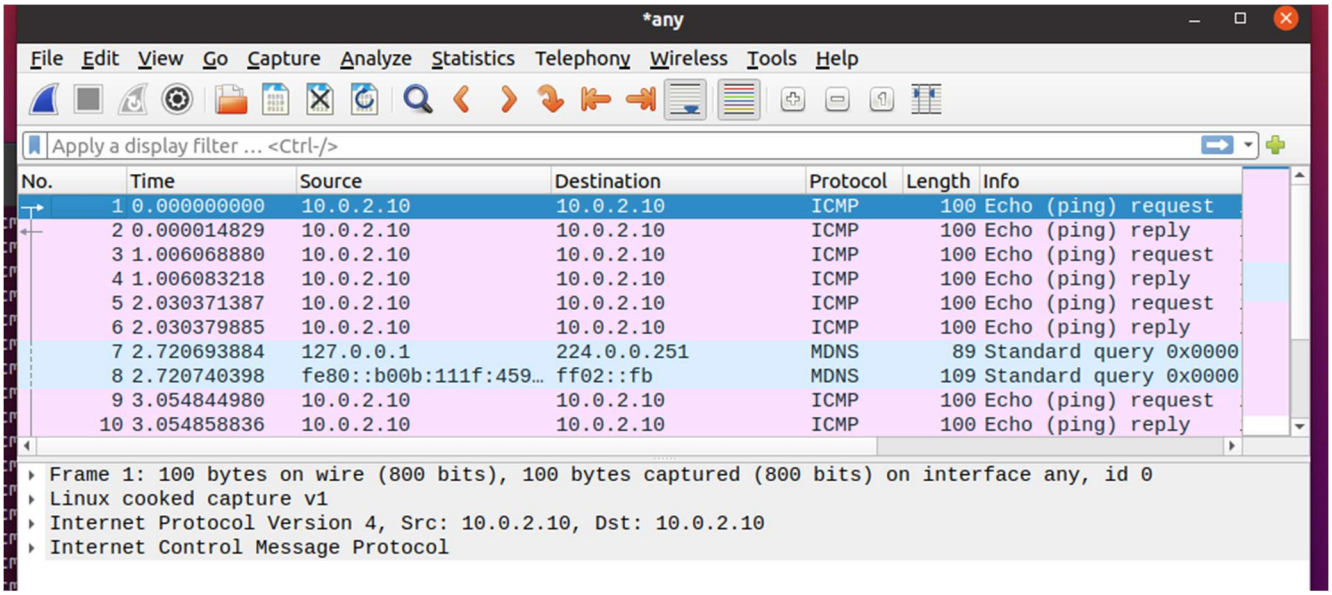
Task 2: Ping PDU (Packet Data Units or Packets) Capture

Step 1: Assign an IP address to the system (Host).

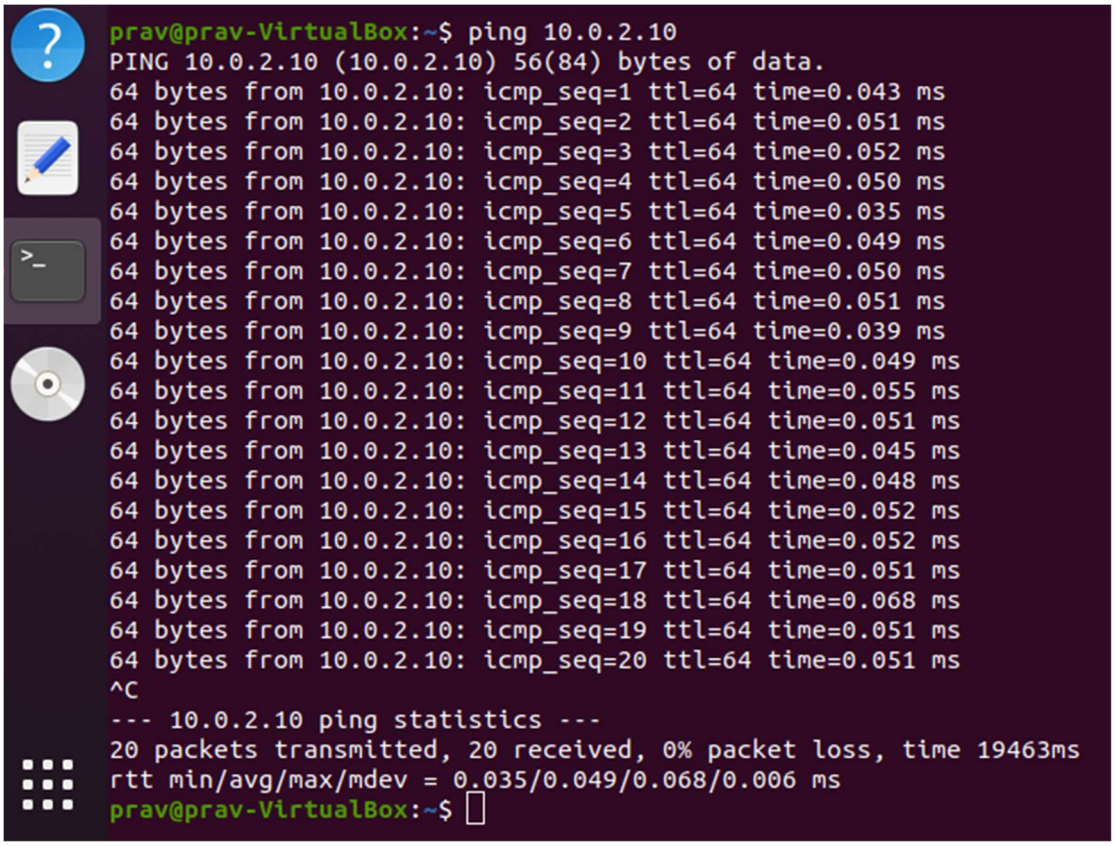
sudo ifconfig enp0s3 10.0.2.10 netmask 255.255.255.0



Step 2: Launch Wireshark and select ‘any’ interface



Step 3: In terminal, type **ping 10.0.your_section.your_sno**



Observations to be made

Step 4: Analyze the following in Terminal

- TTL = 64
- Protocol used by ping = ICMP
- Time -:

  Min = 0.035ms

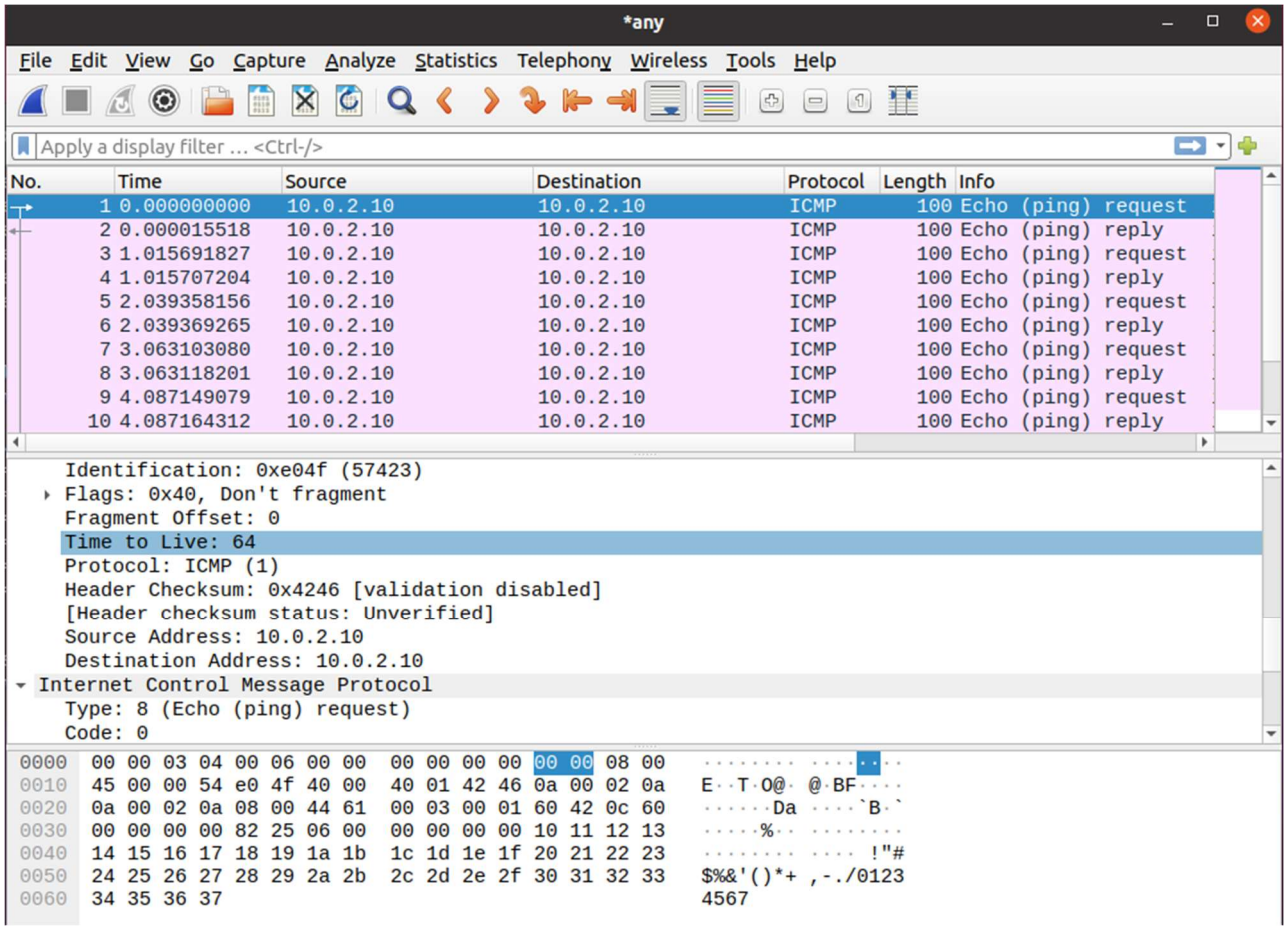
  Avg = 0.049ms

  Max = 0.068ms

  Mdev = 0.006ms

Step 5: Analyze the following in Wireshark

On Packet List Pane, select the first echo packet on the list. On Packet Details Pane, click on each of the four "+" to expand the information. Analyze the frames with the first echo request and echo reply and complete the table below.



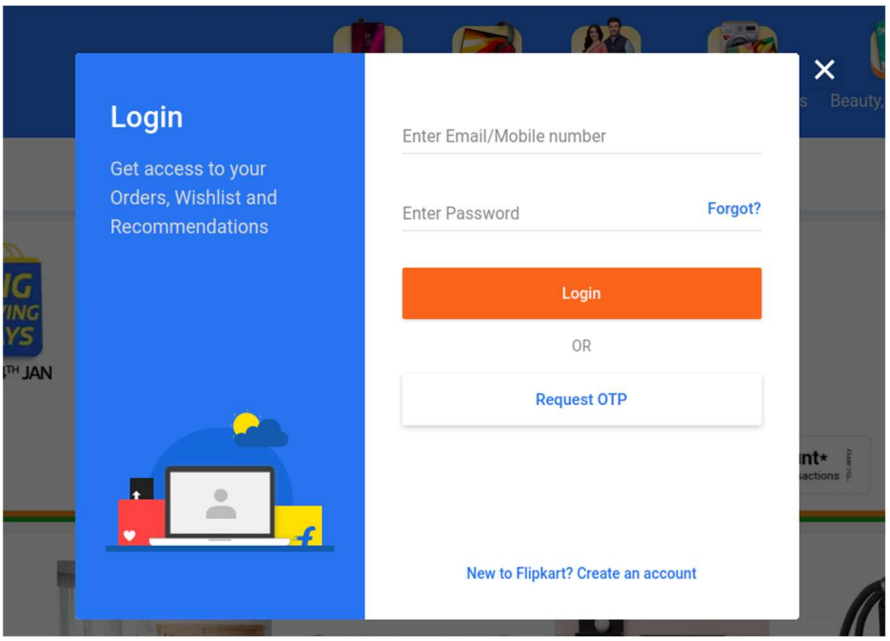
Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.2.10	10.0.2.10
Destination IP address	10.0.2.10	10.0.2.10
ICMP Type Value	8(Echo (ping) request)	0(Echo (ping) reply)
ICMP Code Value	0	0
Source Ethernet Address	00:00:00:00	00:00:00:00
Destination Ethernet Address	00:00:00:00	00:00:00:00
Internet Protocol Version	4	4
Time To Live (TTL) Value	64	64

Task 3: HTTP PDU Capture

Using Wireshark's Filter feature

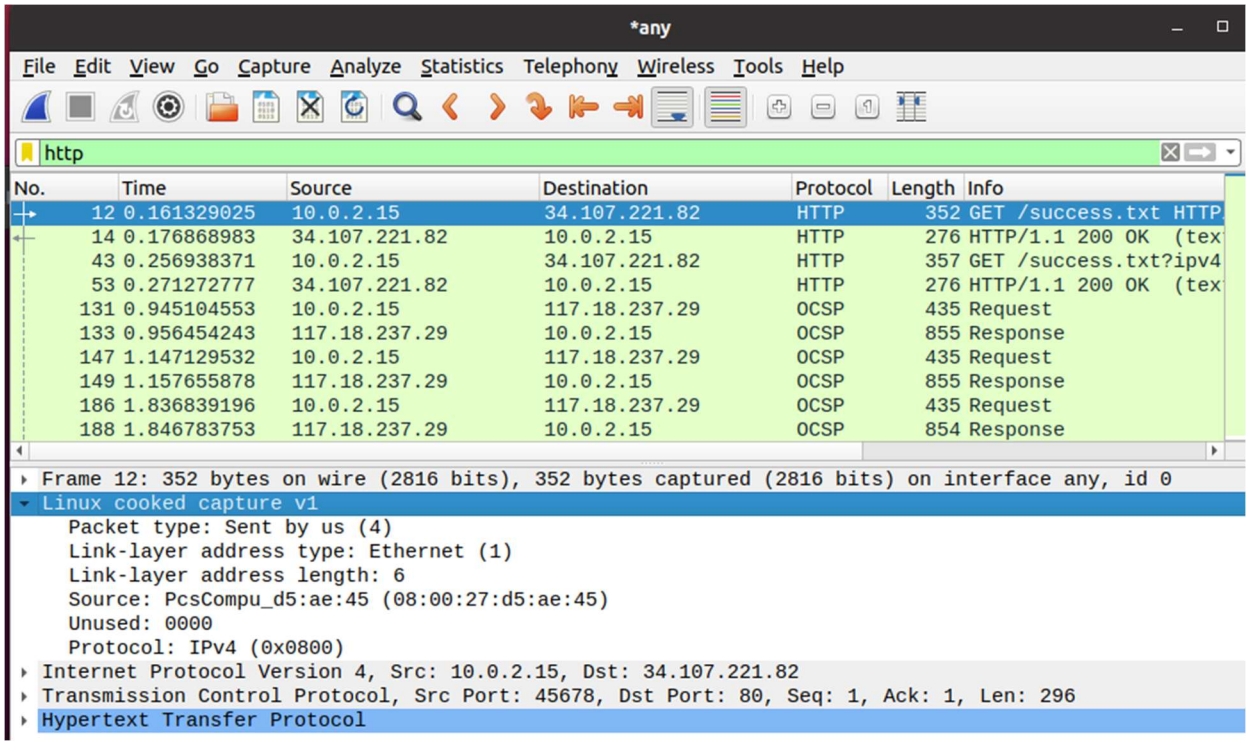
Step 1: Launch Wireshark and select 'any' interface. On the Filter toolbar, type-in 'http' and press enter

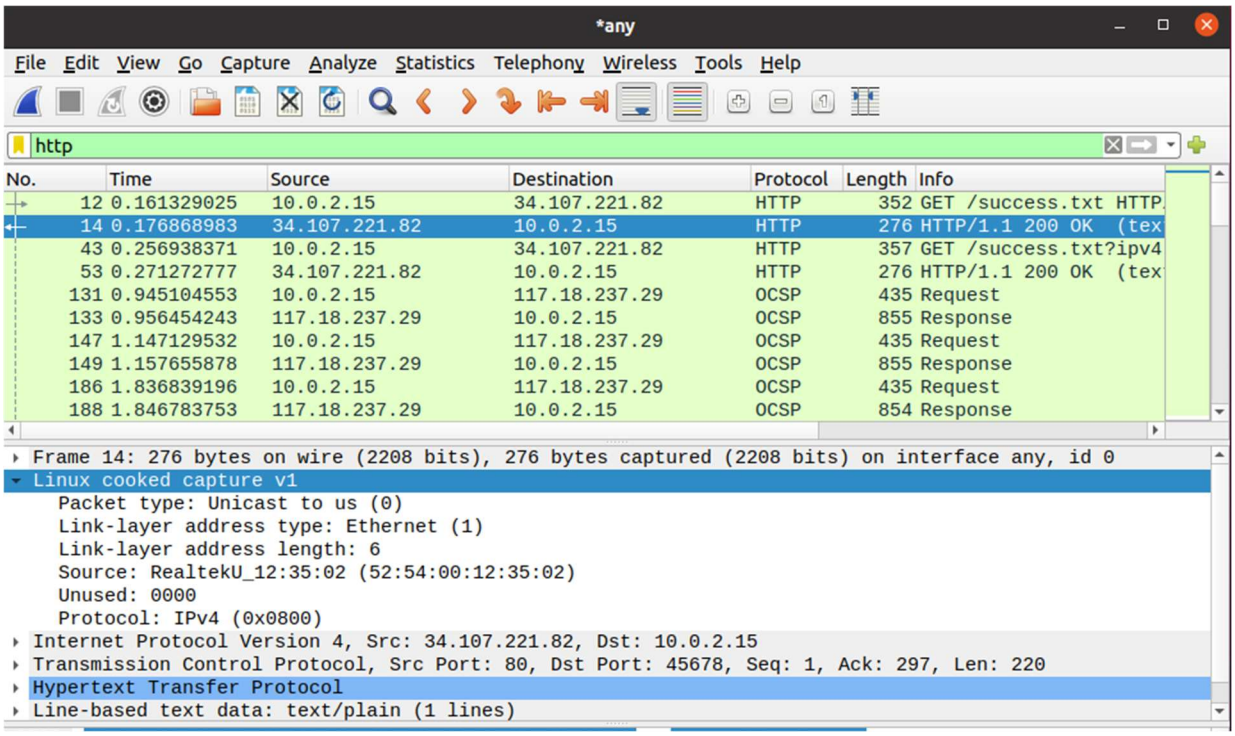
Step 2: Open Firefox browser, and browse www.flipkart.com



Observations to be made

Step 3: Analyze the first (interaction of host to the web server) and second frame (response of server to the client). By analyzing the filtered frames, complete the table below:

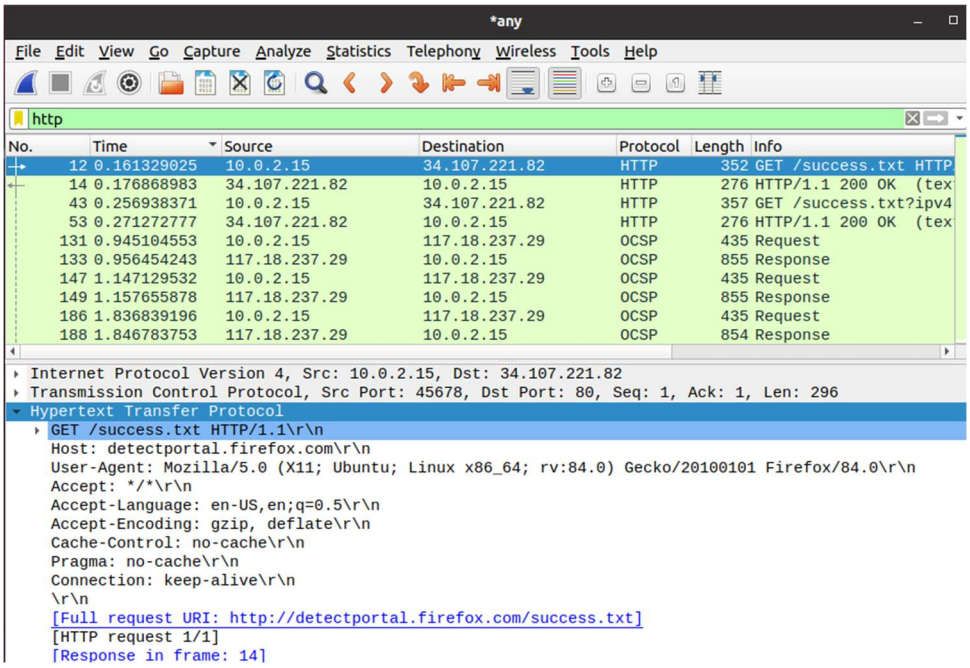


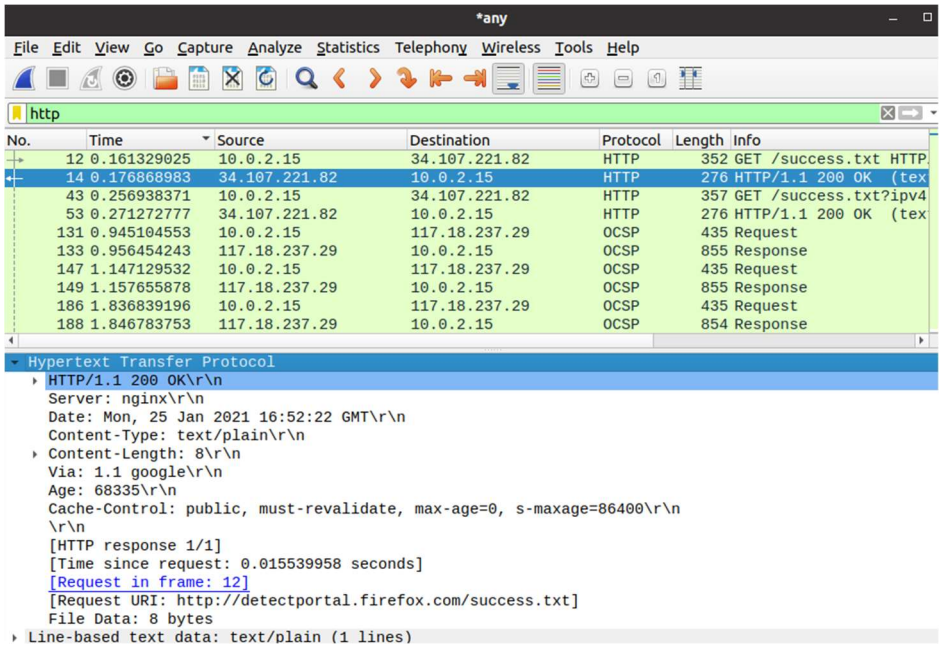


NOTE -: Here the system IP address is 10.0.2.15

Details	First Echo Request	First Echo Reply
Frame Number	12	14
Source Port	45678	80
Destination Port	80	45678
Source IP address	10.0.2.15	34.107.221.82
Destination IP address	34.107.221.82	10.0.2.15
Source Ethernet Address	PcsCompu_d5:ae:45 08:00:27:d5:ae:45	RealtekU_12:35:02 52:54:00:12:35:02

Step 4: Analyze the HTTP request and response and complete the table below.

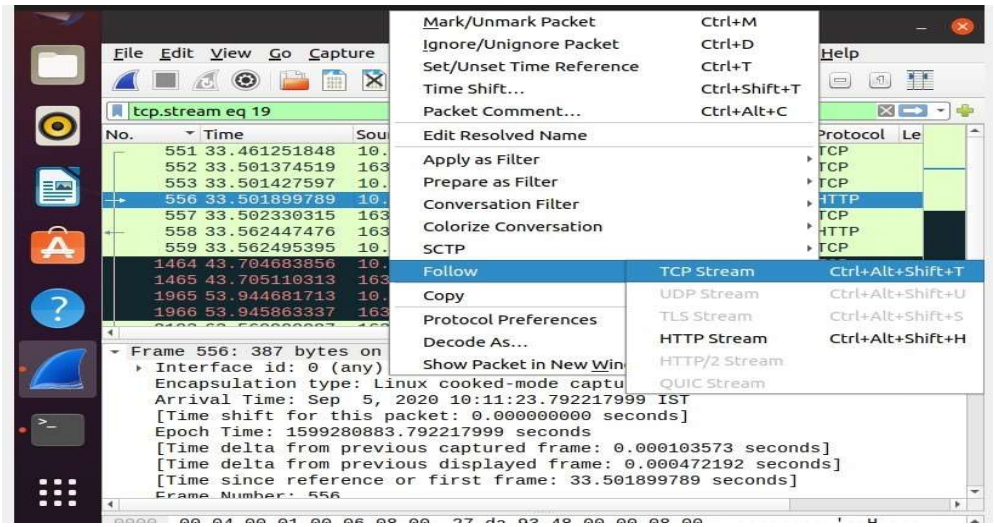




HTTP Request		HTTP Response	
Get	GET/HTTP/1.1	**Server**	nginx\r\n
Host	firefox.com\r\n	**Content-Type**	Text/plain\r\n
User-Agent	Mozilla/5.0(X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0	**Date**	Mon, 25 Jan 2021 16:52:22 GMT\r\n
Accept-Language	en-US, en;q=0.5\r\n	**Location**	https://flipkart.com/\r\n
Accept-Encoding	gzip, deflate\r\n	**Content-Length**	8\r\n
Connection	Keep-alive\r\n	**Connection**	

Using Wireshark's Follow TCP Stream

Step 1: Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'.

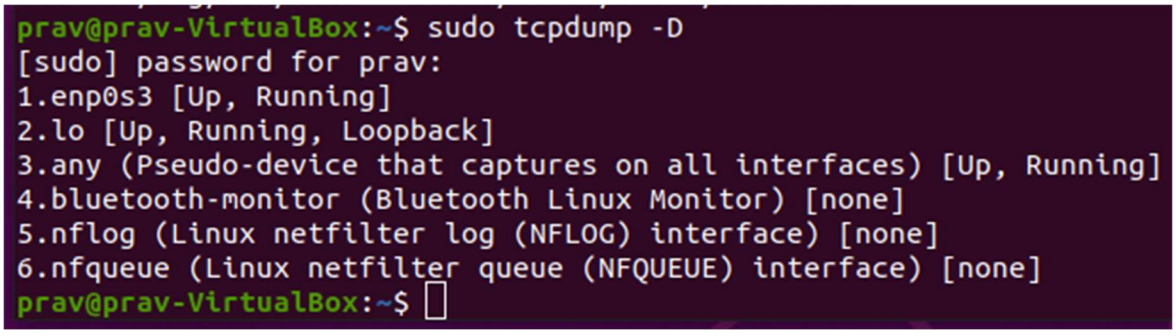


Step 2: Upon following a TCP stream, screenshot the whole window.

Task 4: Capturing packets with tcpdump

Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

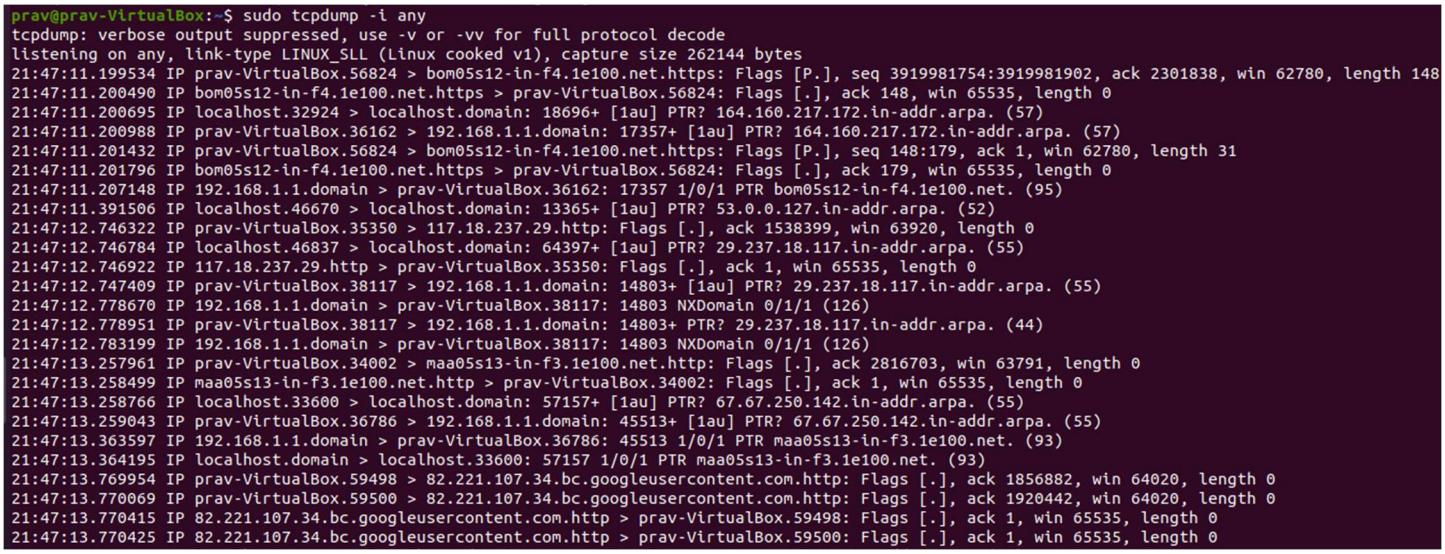
sudo tcpdump -D



Step 2: Capture all packets in any interface by running this command:

sudo tcpdump -i any

Note: Perform some pinging operation while giving above command. Also type www.google.com in browser.



Observations

Step 3: Understand the output format.

- **Timestamp**: The first field is the timestamp of received packet as per the local clock.
- **IP**: Network layer protocol.
- **Source IP>Destination IP**: The next two fields represent the Source IP address followed by Destination IP Address.
- **TCP FLAGS**: Describe the status of the connection. These can also include a combination of values.
- **Sequence Number**: Talks about the sequence number of the data contained in the packet
- **Ack Number**: the sequence number of the next byte the receiver expects to receive.
- **win**: Window Size, which represents the number of bytes available in the receiving buffer.
- **Packet Length**: represents the length, in bytes, of the payload data.

Reference – https://opensource.com/article/18/10/introduction-tcpdump

Step 4: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

sudo tcpdump -i any -c5 icmp

```
prav@prav-VirtualBox:~$ sudo tcpdump -i any -c5 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
22:05:32.776856 IP prav-VirtualBox > bom07s15-in-f14.1e100.net: ICMP echo request, id
6, seq 1, length 64
22:05:32.863730 IP bom07s15-in-f14.1e100.net > prav-VirtualBox: ICMP echo reply, id 6,
 seq 1, length 64
22:05:33.778768 IP prav-VirtualBox > bom07s15-in-f14.1e100.net: ICMP echo request, id
6, seq 2, length 64
22:05:33.815361 IP bom07s15-in-f14.1e100.net > prav-VirtualBox: ICMP echo reply, id 6,
 seq 2, length 64
22:05:34.780550 IP prav-VirtualBox > bom07s15-in-f14.1e100.net: ICMP echo request, id
6, seq 3, length 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
prav@prav-VirtualBox:~$
```

Step 5: Check the packet content. For example, inspect the HTTP content of a web request like this:

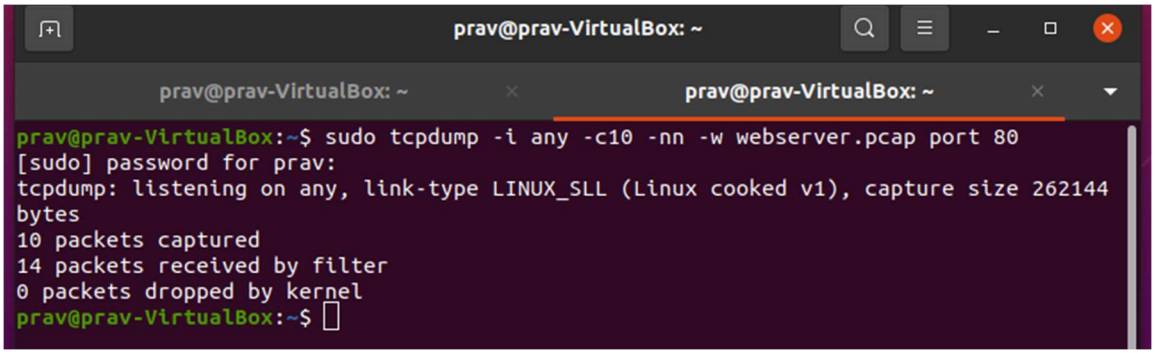
sudo tcpdump -i any -c10 -nn -A port 80

```
prav@prav-VirtualBox:~$ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
22:13:27.454284 IP 10.0.2.15.35512 > 117.18.237.29.80: Flags [S], seq 2579740302, win 64240, options [mss 1460,sackOK,TS val 705535150 ecr 0,nop,wscale 7], length 0
E..<..@.@.      .
...u.....P............nm.........
*...........
22:13:27.485708 IP 10.0.2.15.35514 > 117.18.237.29.80: Flags [S], seq 3301434806, win 64240, options [mss 1460,sackOK,TS val 705535181 ecr 0,nop,wscale 7], length 0
E..<(^@.@...
...u.....P............nm.........
*...........
22:13:27.490242 IP 10.0.2.15.35516 > 117.18.237.29.80: Flags [S], seq 2923110307, win 64240, options [mss 1460,sackOK,TS val 705535186 ecr 0,nop,wscale 7], length 0
E..<..@.@.F.
...u.....P.;..........nm.........
*...........
22:13:27.655946 IP 117.18.237.29.80 > 10.0.2.15.35512: Flags [S.], seq 207040001, ack 2579740303, win 65535, options [mss 1460], length 0
E..,....@...u...
....P...W......`....$........
22:13:27.655990 IP 10.0.2.15.35512 > 117.18.237.29.80: Flags [.], ack 1, win 64240, length 0
E..(..@.@.      .
...u.....P.....W..P...nY..
22:13:27.656470 IP 10.0.2.15.35512 > 117.18.237.29.80: Flags [P.], seq 1:380, ack 1, win 64240, length 379: HTTP: POST / HTTP/1.1
E.....@.@..0
...u.....P.....W..P...o...POST / HTTP/1.1
Host: ocsp.digicert.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 83
Connection: keep-alive
```

```
0Q0O0M0K0I0     ..+........._.z....'.5...C.........a..1a./(..F8.,.......2.....R..I..."U
22:13:27.656901 IP 117.18.237.29.80 > 10.0.2.15.35512: Flags [.], ack 380, win 65535, length 0
E..(....@...u...
....P...W.....
P...*f........
22:13:27.684898 IP 117.18.237.29.80 > 10.0.2.15.35516: Flags [S.], seq 207104001, ack 2923110308, win 65535, options [mss 1460], length 0
E..,....@...u...
....P...X(..;..`.............
22:13:27.684984 IP 10.0.2.15.35516 > 117.18.237.29.80: Flags [.], ack 1, win 64240, length 0
E..(..@.@.F.
...u......P.;...X(.P...nY..
22:13:27.685007 IP 117.18.237.29.80 > 10.0.2.15.35514: Flags [S.], seq 207168001, ack 3301434807, win 65535, options [mss 1460], length 0
E..,....@...u...
....P...Y".....`.............
10 packets captured
11 packets received by filter
0 packets dropped by kernel
prav@prav-VirtualBox:~$
```

Step 6: To save packets to file instead of displaying them on screen, use the option -w:

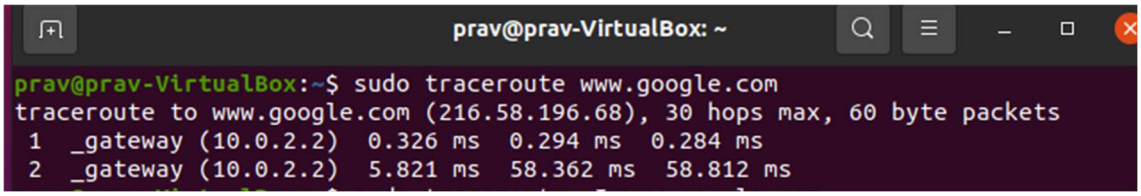
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80



Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com



Step 2: Analyze destination address of google.com and no. of hops

Destination address = 216.58.196.68

No. of hops = 30

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the *-n* option

sudo traceroute -n www.google.com

```
prav@prav-VirtualBox:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (172.217.160.164), 30 hops max, 60 byte packets
 1  10.0.2.2  0.148 ms  0.120 ms  0.109 ms
 2  10.0.2.2  241.414 ms  241.403 ms  242.208 ms
prav@prav-VirtualBox:~$
```

Compared to previous step this has taken lesser time.

Step 4: The -I option is necessary so that the traceroute uses ICMP.

sudo traceroute -I www.google.com

```
prav@prav-VirtualBox:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (216.58.196.68), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.303 ms  0.241 ms  0.222 ms
 2  192.168.1.1 (192.168.1.1)  57.007 ms * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  182.79.177.69 (182.79.177.69)  17.019 ms  17.919 ms  18.331 ms
 8  72.14.208.234 (72.14.208.234)  23.993 ms  23.139 ms  23.079 ms
 9  108.170.253.97 (108.170.253.97)  23.057 ms  22.974 ms  22.957 ms
10  108.170.253.104 (108.170.253.104)  18.229 ms  18.213 ms  18.007 ms
11  64.233.174.3 (64.233.174.3)  21.510 ms  21.038 ms  22.536 ms
12  209.85.251.242 (209.85.251.242)  41.061 ms  41.034 ms  40.028 ms
13  108.170.248.177 (108.170.248.177)  38.645 ms  39.680 ms  36.777 ms
14  209.85.255.209 (209.85.255.209)  36.231 ms  35.308 ms  38.310 ms
15  bom05s11-in-f4.1e100.net (216.58.196.68)  38.295 ms  38.711 ms  38.634 ms
prav@prav-VirtualBox:~$
```

No. of hops =15

Step 5: By default, traceroute uses icmp (ping) packets. If you'd rather test a TCP connection to gather data more relevant to web server, you can use the -T flag.

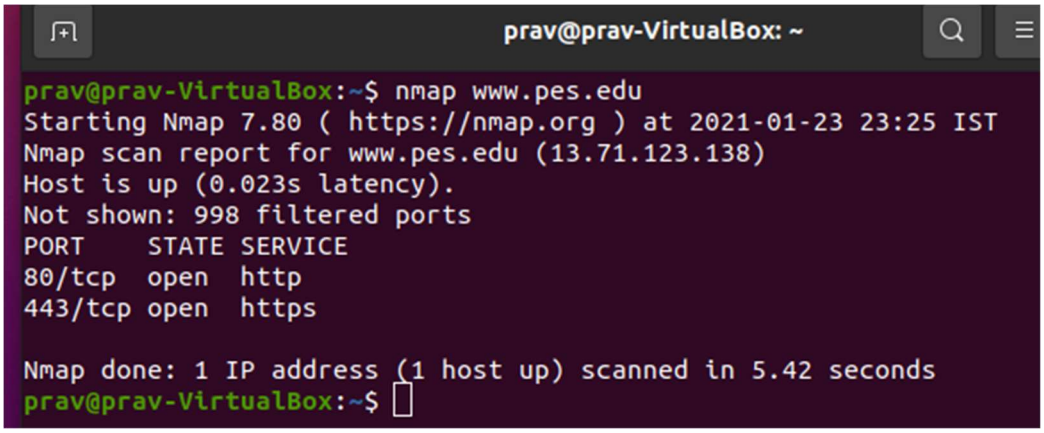
sudo traceroute -T www.google.com

```
prav@prav-VirtualBox: ~
prav@prav-VirtualBox:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (172.217.167.132), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.240 ms  0.204 ms  0.190 ms
 2  maa03s26-in-f4.1e100.net (172.217.167.132)  67.169 ms  82.962 ms  69.869 ms
prav@prav-VirtualBox:~$
```

Task 6: Explore an entire network for information (Nmap)

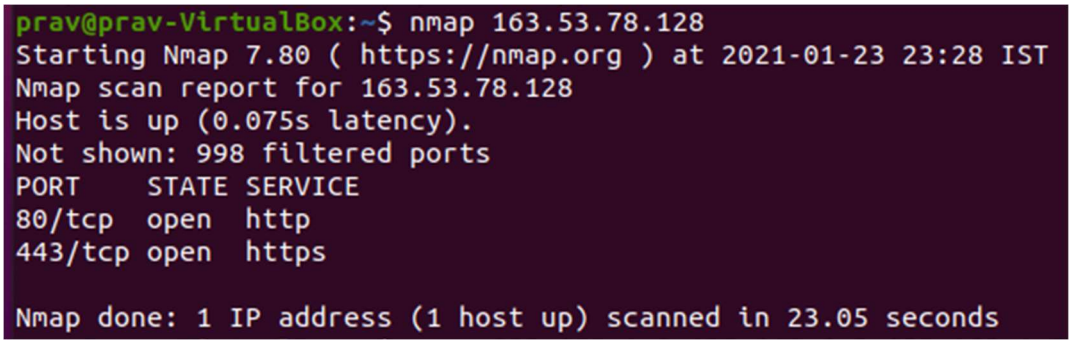
Step 1: You can scan a host using its host name or IP address, for instance.

nmap www.pes.edu



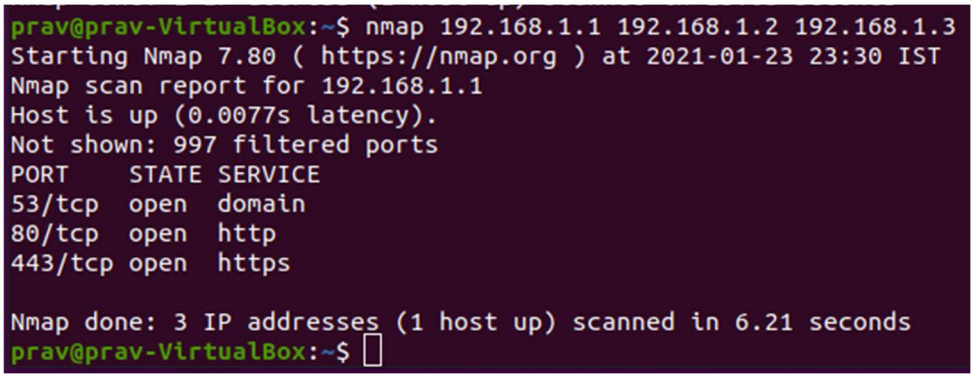
Step 2: Alternatively, use an IP address to scan.

nmap 163.53.78.128



Step 3: Scan multiple IP address or subnet (IPv4)

nmap 192.168.1.1 192.168.1.2 192.168.1.3



Task 7 a): Netcat as Chat tool

a) Intra system communication (Using 2 terminals in the same system)

Step 1: Open a terminal (Ctrl+Alt+T). This will act as a Server.

Step 2: Type **nc -l any_portnum** (For eg., nc -l 1234). It will go to listening mode.

Step 3: Open another terminal and this will act as a client.

Step 4: Type nc <your-system-ip-address> portnum

Note: portnum should be same in both the terminals (for eg., nc 10.0.2.8 1234)

Step 5: Type anything in client will appear in server