

Week #1

Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute.

Learn and Understand Network Tools

1. Wireshark

- ☐ Perform and analyze Ping PDU capture
- ☐ Examine HTTP packet capture
- ☐ Analyze HTTP packet capture using filter

2. Tcpdump

- Capture packets

3. Ping

- Test the connectivity between 2 systems

4. Traceroute

- Perform traceroute checks

5. Nmap

- Explore an entire network

IMPORTANT INSTRUCTIONS:

- This manual is written for Ubuntu Linux OS only. You can also execute these experiments on VirtualBox or VMWare platform.
- For few tasks, you may need to create 2 VMs for experimental setup.
- Perform **sudo apt-get update** before installing any tool or utility.
- Install any tool or utility using the command **sudo apt-get install name_of_the_tool**
- Take screenshots wherever necessary and upload it to Edmodo as a single PDF file. (Refer general guidelines for submission requirements).
- To define an IP address for your machine (e.g., Section – ‘a’ & Serial number is 1, then your IP address should be 10.0.1.1. Section – ‘h’ & Serial number is 23, then your IP address should be 10.0.8.23) – applicable only for relevant tasks (which doesn't requires internet connectivity to execute the tasks).

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 following table:

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address	
enp0s3	10.0.2.15	08:00:27:66:f4:3e	
lo	127.0.0.1	-	

```
vishwas@pop-os: ~  
vishwas@pop-os:~$ 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::726e:81b:e051:d778 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:66:f4:3e txqueuelen 1000 (Ethernet)  
    RX packets 291712 bytes 431464970 (431.4 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 66646 bytes 4562908 (4.5 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 1062 bytes 1369797 (1.3 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1062 bytes 1369797 (1.3 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vishwas@pop-os:~$
```

The hardware address and the IP address is mentioned,when ifconfig is typed in the terminal.

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

sudo ifconfig interface_name 10.0.your_section.your_sno netmask 255.255.255.0 (or)

sudo ip addr add 10.0.your_section.your_sno /24 dev interface_name

```
vishwas@pop-os: ~  
vishwas@pop-os:~$ sudo ifconfig enp0s3 10.0.6.57 netmask 255.255.255.0  
vishwas@pop-os:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.6.57 netmask 255.255.255.0 broadcast 10.0.6.255  
    inet6 fe80::726e:81b:e051:d778 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:66:f4:3e txqueuelen 1000 (Ethernet)  
    RX packets 319427 bytes 460821369 (460.8 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 81418 bytes 9939154 (9.9 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 7307 bytes 1934469 (1.9 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 7307 bytes 1934469 (1.9 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vishwas@pop-os:~$
```

10.0.6.57 is assigned as the IP address to the interface.

Step 3: To activate / deactivate a network interface, type.

sudo ifconfig interface_name down

sudo ifconfig interface_name up

```
vishwas@pop-os:~$ sudo ifconfig enp0s3 up
```

The configured interface is set to

up and running if it isn't.

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

ip neigh

```
vishwas@pop-os:~$ ip neigh  
192.168.0.1 dev enp0s3 FAILED  
vishwas@pop-os:~$ ip neigh  
192.168.0.1 dev enp0s3 INCOMPLETE  
vishwas@pop-os:~$ ip neigh  
192.168.0.1 dev enp0s3 INCOMPLETE  
vishwas@pop-os:~$ ip neigh  
192.168.0.1 dev enp0s3 INCOMPLETE  
vishwas@pop-os:~$ ip neigh  
192.168.0.1 dev enp0s3 INCOMPLETE  
vishwas@pop-os:~$
```

The neighbor table is shown in the output.

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

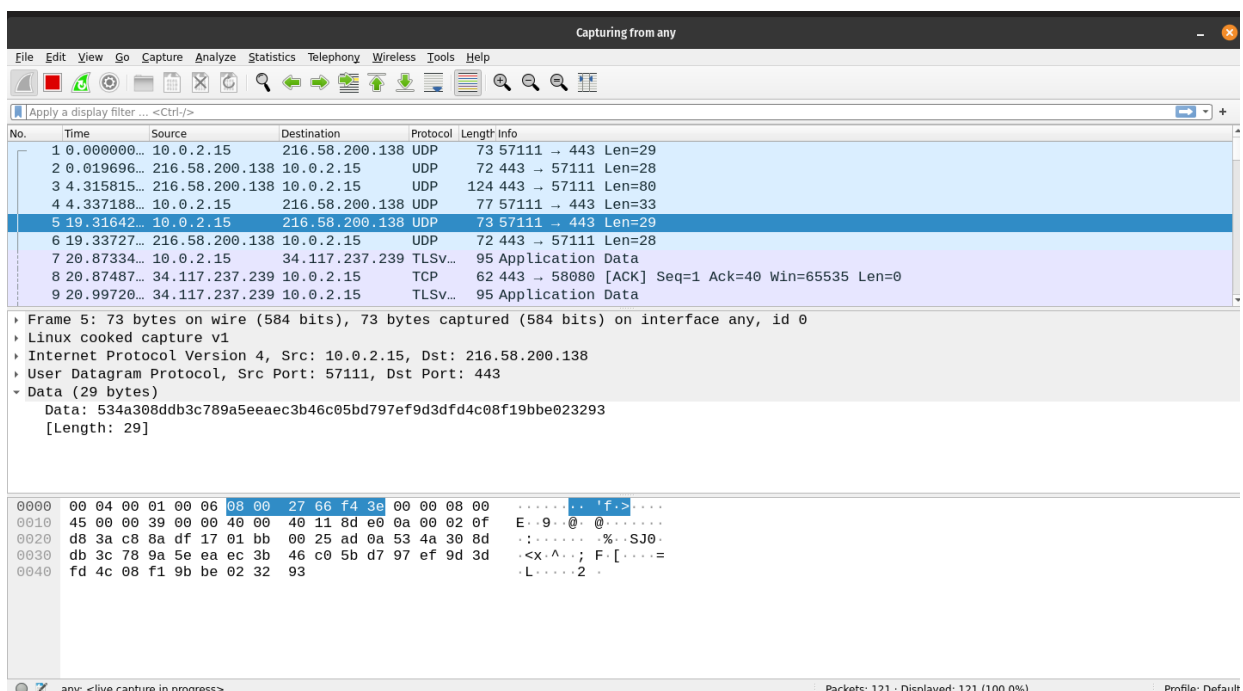
Step 1: Assign an IP address to the system (Host).

Note: IP address of your system should be 10.0.your_section.your_sno.

```
vishwas@pop-os: ~  
vishwas@pop-os:~$ sudo ifconfig enp0s3 10.0.6.57 netmask 255.255.255.0  
vishwas@pop-os:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.6.57 netmask 255.255.255.0 broadcast 10.0.6.255  
    inet6 fe80::726e:81b:e051:d778 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:66:f4:3e txqueuelen 1000 (Ethernet)  
    RX packets 319427 bytes 460821369 (460.8 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 81418 bytes 9939154 (9.9 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 7307 bytes 1934469 (1.9 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 7307 bytes 1934469 (1.9 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vishwas@pop-os:~$
```

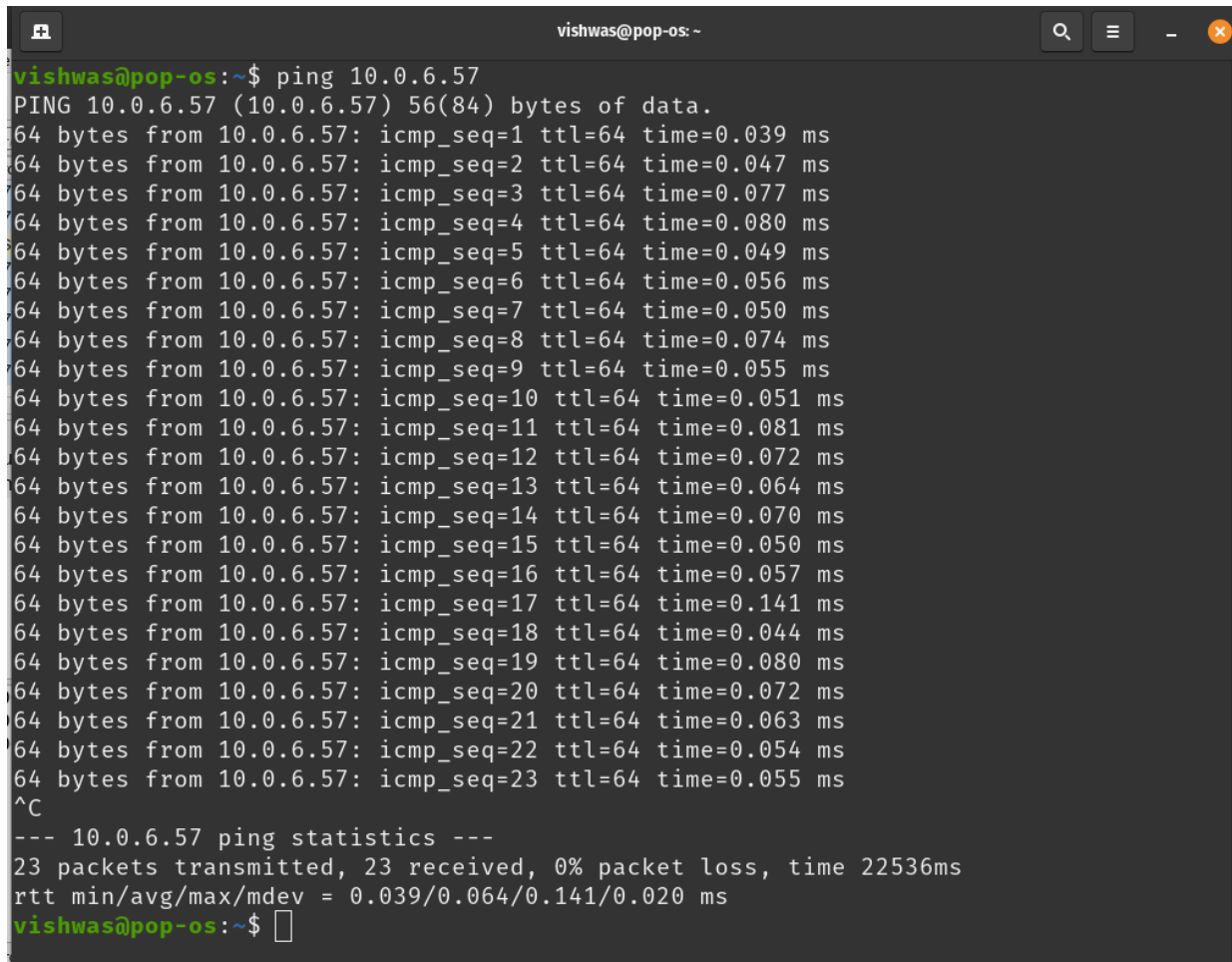
The IP address is set to 10.0.6.57.

Step 2: Launch Wireshark and select ‘any’ interfa



Wireshark on launch and opened into “any”.

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



```
vishwas@pop-os:~$ ping 10.0.6.57
PING 10.0.6.57 (10.0.6.57) 56(84) bytes of data.
64 bytes from 10.0.6.57: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 10.0.6.57: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 10.0.6.57: icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from 10.0.6.57: icmp_seq=4 ttl=64 time=0.080 ms
64 bytes from 10.0.6.57: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.6.57: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.0.6.57: icmp_seq=7 ttl=64 time=0.050 ms
64 bytes from 10.0.6.57: icmp_seq=8 ttl=64 time=0.074 ms
64 bytes from 10.0.6.57: icmp_seq=9 ttl=64 time=0.055 ms
64 bytes from 10.0.6.57: icmp_seq=10 ttl=64 time=0.051 ms
64 bytes from 10.0.6.57: icmp_seq=11 ttl=64 time=0.081 ms
64 bytes from 10.0.6.57: icmp_seq=12 ttl=64 time=0.072 ms
64 bytes from 10.0.6.57: icmp_seq=13 ttl=64 time=0.064 ms
64 bytes from 10.0.6.57: icmp_seq=14 ttl=64 time=0.070 ms
64 bytes from 10.0.6.57: icmp_seq=15 ttl=64 time=0.050 ms
64 bytes from 10.0.6.57: icmp_seq=16 ttl=64 time=0.057 ms
64 bytes from 10.0.6.57: icmp_seq=17 ttl=64 time=0.141 ms
64 bytes from 10.0.6.57: icmp_seq=18 ttl=64 time=0.044 ms
64 bytes from 10.0.6.57: icmp_seq=19 ttl=64 time=0.080 ms
64 bytes from 10.0.6.57: icmp_seq=20 ttl=64 time=0.072 ms
64 bytes from 10.0.6.57: icmp_seq=21 ttl=64 time=0.063 ms
64 bytes from 10.0.6.57: icmp_seq=22 ttl=64 time=0.054 ms
64 bytes from 10.0.6.57: icmp_seq=23 ttl=64 time=0.055 ms
^C
--- 10.0.6.57 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22536ms
rtt min/avg/max/mdev = 0.039/0.064/0.141/0.020 ms
vishwas@pop-os:~$
```

Observations to be made

Step 4: Analyze the following in Terminal

- TTL
- Protocol used by ping
- Time

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	207	208
Source IP address	10.0.6.57	10.0.6.57
Destination IP address	10.0.6.57	10.0.6.57
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00

NAME:VISHWAS M
SRN:PES2UG20CS390

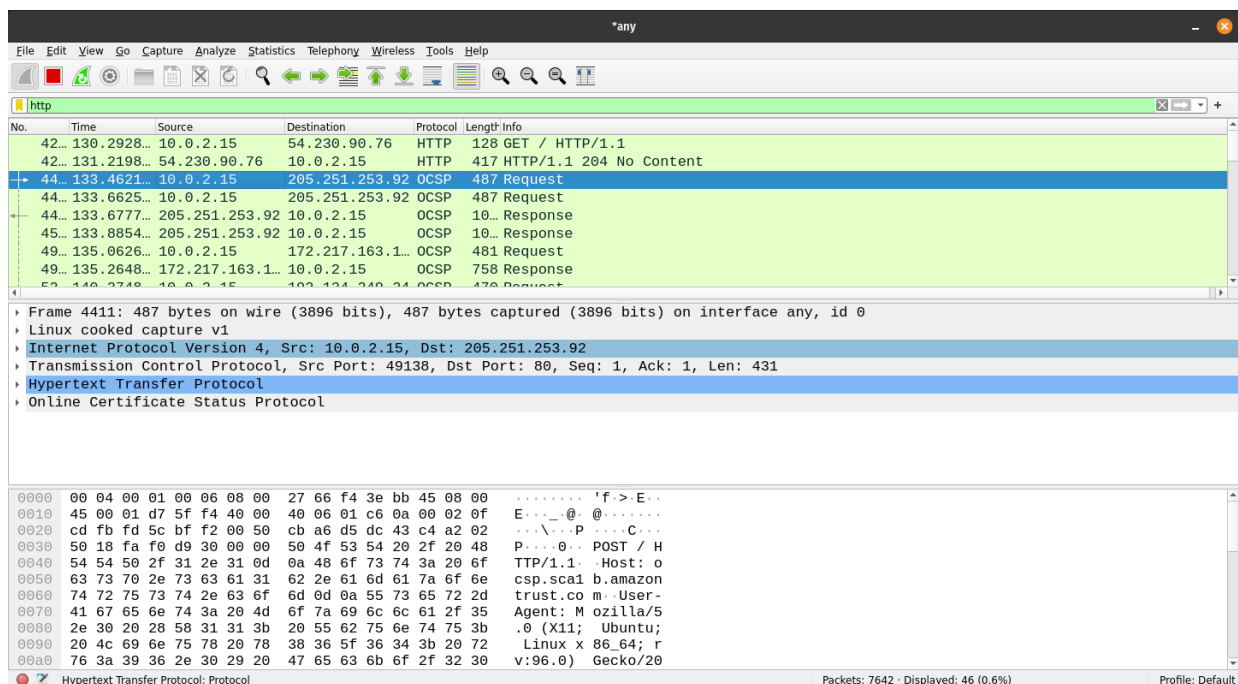
SEC:F
SUB: CN LAB

Destination Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
Internet Protocol Version	4	4
Time To Live (TTL) Value	64(reply in 208)	64(request in 207)

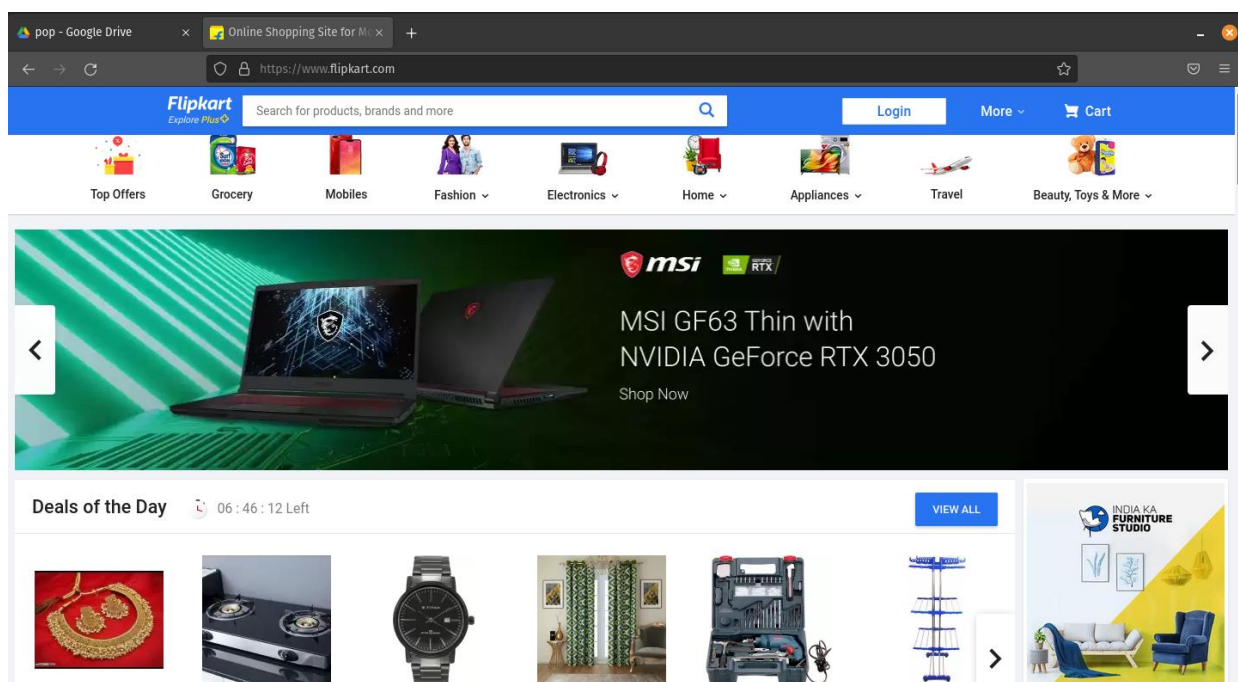
Task 3: HTTP PDU Capture

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:

Details	First Echo Request	First Echo Reply
Frame Number	4411	4470
Source Port	49138	80
Destination Port	80	49138
Source IP address	10.0.2.15	205.251.253.92
Destination IP address	205.251.253.92	10.0.2.15
Source Ethernet Address	08:00:27:66:f4:3e	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	08:00:27:66:f4:3e

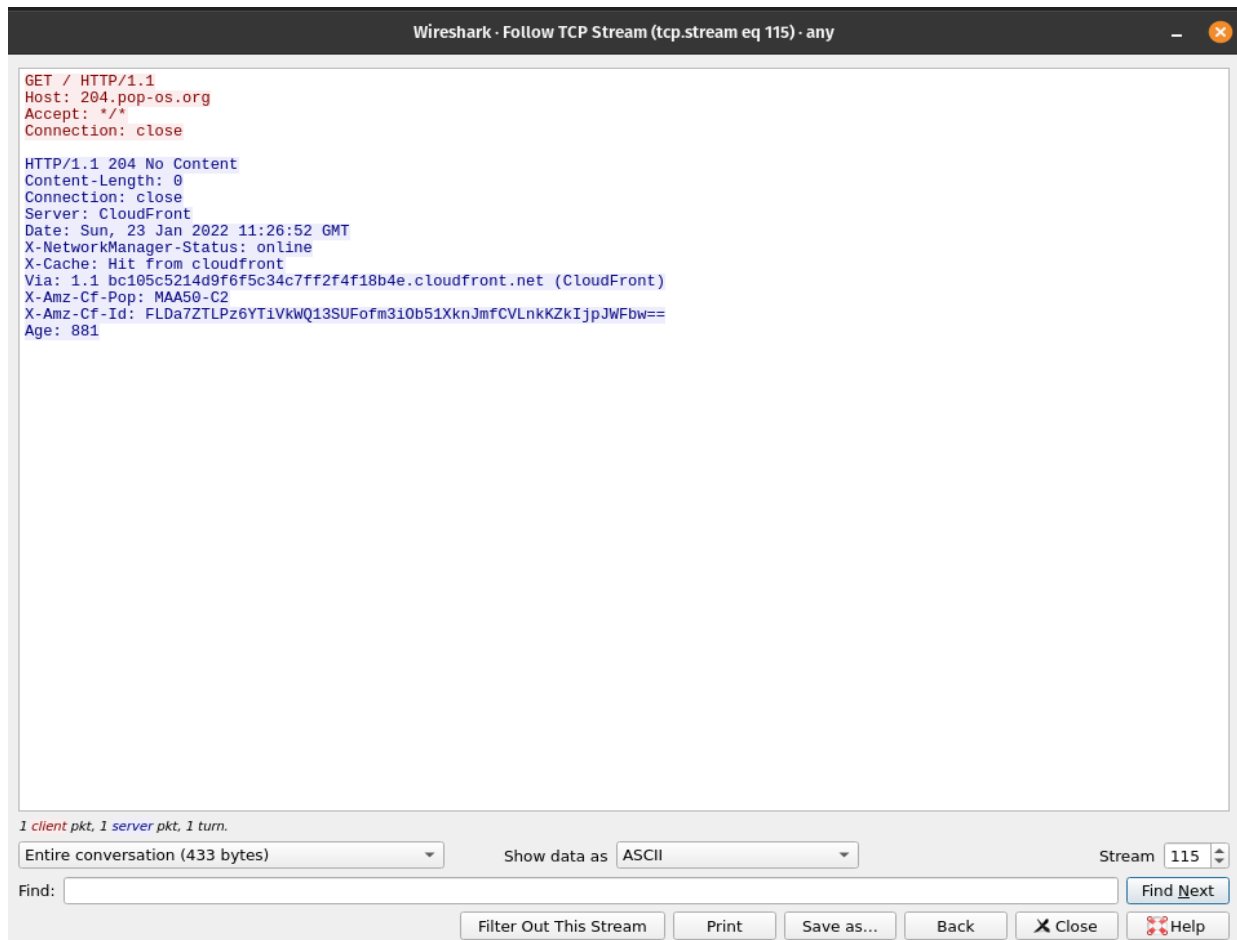
Step 4: Analyze the HTTP request and response and complete the table below.

HTTP Request		HTTP Response	
Get	/HTTP/1.1	Server	ECS (oxr/8323)
Host	Ocsp.scalb.amazontrust.com	Content-Type	Application/ocsp-request
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0	Date	Sun,23 Jan 2022 11:41:36 GMT
Accept-Language	En-US,en;q=0.5	Location	<NOT SPECIFIED>
Accept-Encoding	Qzip, deflate	Content-Length	471
Connection	Keep-alive	Connection	Keep-alive

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'. For demo purpose, a packet containing the HTTP GET request "GET / HTTP / 1.1" can be selected.

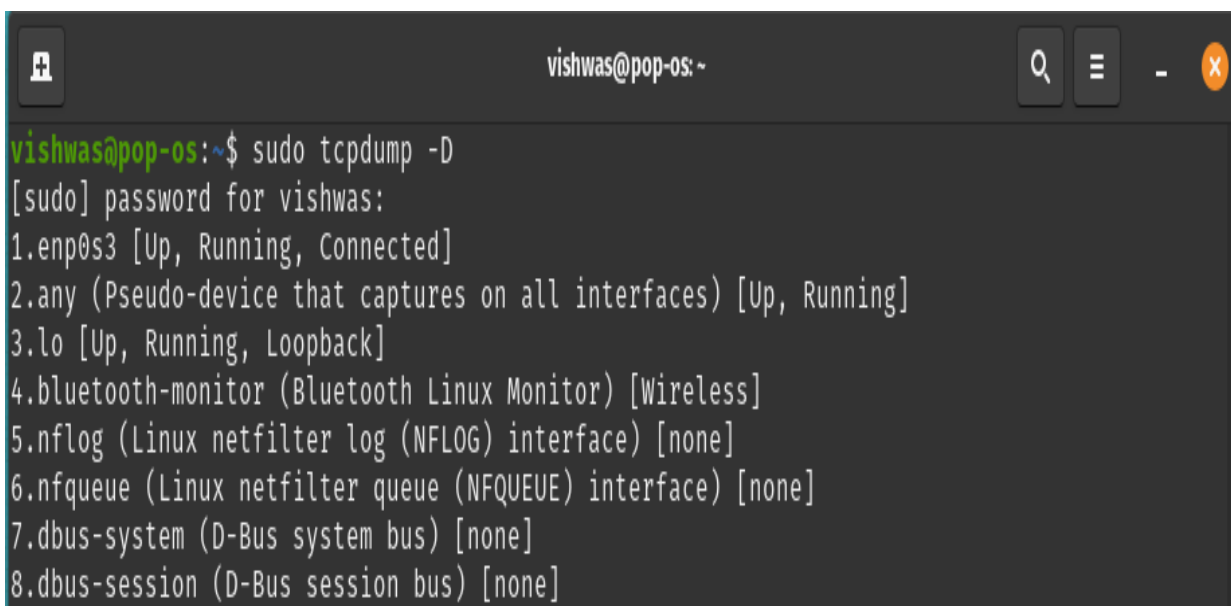
Step 2: Upon following a TCP stream, screenshot the whole window.



Task 4: Capturing packets with tcpdump

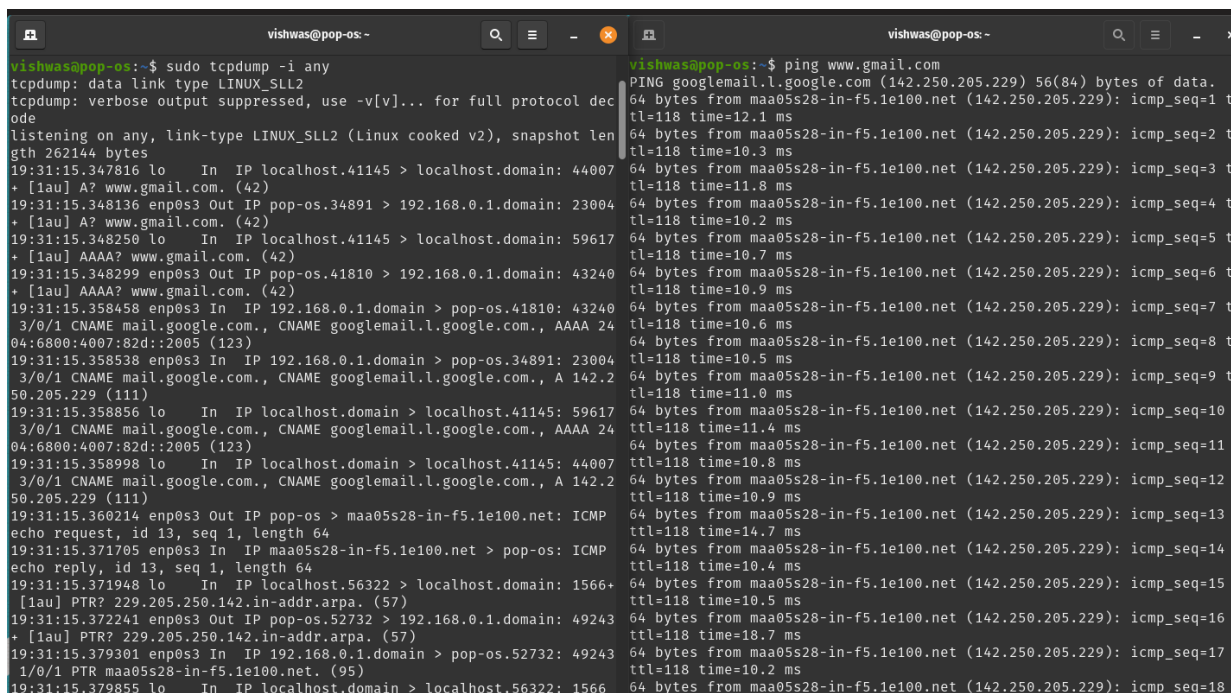
Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

sudo tcpdump -D



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

sudo tcpdump -i any



The image shows two terminal windows from a user named 'vishwas' on a 'pop-os' system. The left window shows the output of the command 'sudo tcpdump -i any', which captures network traffic on all interfaces. The output shows various packets, including ARP requests, ICMP echo requests (ping), and DNS queries. The right window shows the output of the command 'ping www.gmail.com', which displays the results of a ping operation to the IP address 142.250.205.229. The output shows 18 successful ping attempts, each with a 64-byte payload and a response time between 10.2 ms and 11.8 ms.

```
vishwas@pop-os:~$ sudo tcpdump -i any
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decoding
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
19:31:15.347816 lo In IP localhost.41145 > localhost.domain: 44007
* [1au] A? www.gmail.com. (42)
19:31:15.348136 enp0s3 Out IP pop-os.34891 > 192.168.0.1.domain: 23004
* [1au] A? www.gmail.com. (42)
19:31:15.348250 lo In IP localhost.41145 > localhost.domain: 59617
* [1au] AAAA? www.gmail.com. (42)
19:31:15.348299 enp0s3 Out IP pop-os.41810 > 192.168.0.1.domain: 43240
* [1au] AAAA? www.gmail.com. (42)
19:31:15.358458 enp0s3 In IP 192.168.0.1.domain > pop-os.41810: 43240
3/0/1 CNAME mail.google.com., CNAME googlemail.l.google.com., AAAA 24
04:6800:4007:82d::2005 (123)
19:31:15.358538 enp0s3 In IP 192.168.0.1.domain > pop-os.34891: 23004
3/0/1 CNAME mail.google.com., CNAME googlemail.l.google.com., A 142.2
50.205.229 (111)
19:31:15.358856 lo In IP localhost.domain > localhost.41145: 59617
3/0/1 CNAME mail.google.com., CNAME googlemail.l.google.com., AAAA 24
04:6800:4007:82d::2005 (123)
19:31:15.358998 lo In IP localhost.domain > localhost.41145: 44007
3/0/1 CNAME mail.google.com., CNAME googlemail.l.google.com., A 142.2
50.205.229 (111)
19:31:15.360214 enp0s3 Out IP pop-os > maa05s28-in-f5.1e100.net: ICMP
echo request, id 13, seq 1, length 64
19:31:15.371705 enp0s3 In IP maa05s28-in-f5.1e100.net > pop-os: ICMP
echo reply, id 13, seq 1, length 64
19:31:15.371948 lo In IP localhost.56322 > localhost.domain: 1566+
* [1au] PTR? 229.205.250.142.in-addr.arpa. (57)
19:31:15.372241 enp0s3 Out IP pop-os.52732 > 192.168.0.1.domain: 49243
* [1au] PTR? 229.205.250.142.in-addr.arpa. (57)
19:31:15.379301 enp0s3 In IP 192.168.0.1.domain > pop-os.52732: 49243
1/0/1 PTR maa05s28-in-f5.1e100.net. (95)
19:31:15.379855 lo In IP localhost.domain > localhost.56322: 1566

vishwas@pop-os:~$ ping www.gmail.com
PING googlemail.l.google.com (142.250.205.229) 56(84) bytes of data.
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=1 t
tl=118 time=12.1 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=2 t
tl=118 time=10.3 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=3 t
tl=118 time=11.8 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=4 t
tl=118 time=10.2 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=5 t
tl=118 time=10.7 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=6 t
tl=118 time=10.9 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=7 t
tl=118 time=10.6 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=8 t
tl=118 time=10.5 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=9 t
tl=118 time=11.0 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=10
ttl=118 time=11.4 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=11
ttl=118 time=10.8 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=12
ttl=118 time=10.9 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=13
ttl=118 time=14.7 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=14
ttl=118 time=10.4 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=15
ttl=118 time=10.5 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=16
ttl=118 time=18.7 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=17
ttl=118 time=10.2 ms
64 bytes from maa05s28-in-f5.1e100.net (142.250.205.229): icmp_seq=18
```

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

Observation

Step 3: Understand the output format.

The above command is used to capture all the packets from all the interfaces. ICMP, UDP and TCP are the main packets that are visible in the above screenshot. The timestamp followed by the link level headers, then by ARP/RARP packets if any, Then by IPv4 packets if any, followed by TCP packets. The sequence numbers and the length finish defining the outputs.

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

```
vishwas@pop-os:~$ sudo tcpdump -i any -c5 icmp
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol de
code
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot le
ngth 262144 bytes
19:49:33.259255 enp0s3 Out IP pop-os > media-router-fp71.canary.media
.vip.sg3.yahoo.com: ICMP echo request, id 14, seq 1, length 64
19:49:33.302850 enp0s3 In  IP media-router-fp71.canary.media.vip.sg3.
yahoo.com > pop-os: ICMP echo reply, id 14, seq 1, length 64
19:49:34.275888 enp0s3 Out IP pop-os > media-router-fp71.canary.media
.vip.sg3.yahoo.com: ICMP echo request, id 14, seq 2, length 64
19:49:34.320429 enp0s3 In  IP media-router-fp71.canary.media.vip.sg3.
yahoo.com > pop-os: ICMP echo reply, id 14, seq 2, length 64
19:49:35.277615 enp0s3 Out IP pop-os > media-router-fp71.canary.media
.vip.sg3.yahoo.com: ICMP echo request, id 14, seq 3, length 64
5 packets captured
6 packets received by filter
0 packets dropped by kernel
vishwas@pop-os:~$
```

```
vishwas@pop-os:~$ ping www.yahoo.com
PING new-fp-shed.wg1.b.yahoo.com (202.165.107.48) 56(84) bytes of data
.
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=1 ttl=49 time=44.2 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=2 ttl=49 time=44.6 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=3 ttl=49 time=44.7 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=4 ttl=49 time=43.6 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=5 ttl=49 time=49.9 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=6 ttl=49 time=44.2 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=7 ttl=49 time=44.6 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=8 ttl=49 time=44.3 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=9 ttl=49 time=43.9 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=10 ttl=49 time=43.6 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=11 ttl=49 time=43.1 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=12 ttl=49 time=46.9 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=13 ttl=49 time=42.8 ms
64 bytes from media-router-fp71.canary.media.vip.sg3.yahoo.com (202.16
5.107.48): icmp_seq=14 ttl=49 time=47.2 ms
^C
--- new-fp-shed.wg1.b.yahoo.com ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13072ms
rtt min/avg/max/mdev = 42.768/44.835/49.852/1.847 ms
vishwas@pop-os:~$
```

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

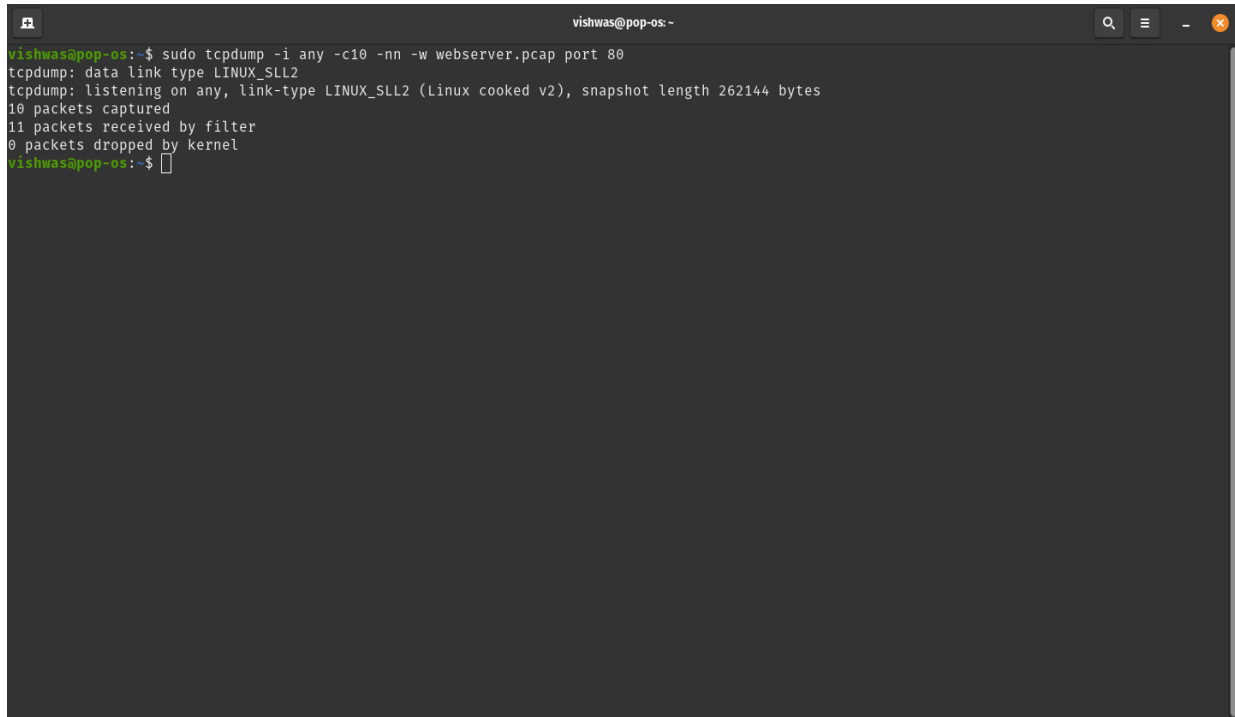
sudo tcpdump -i any -c10 -nn -A port 80

```
vishwas@pop-os:~$ sudo tcpdump -i any -c5 -A port 80
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol de
code
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
19:55:39.110758 enp0s3 Out IP pop-os.51196 > 103.16.70.139.http: Flags [S], seq 2577378844, win 64240, options [mss 1460,sackOK,TS val 4007744
84 ecr 0,nop,wscale 7], length 0
E..<..@.@...
...g.F....P.....
..UT.....
19:55:39.121645 enp0s3 In  IP 103.16.70.139.http > pop-os.51196: Flags [S.], seq 1679424001, ack 2577378845, win 65535, options [mss 1460], le
ngth 0
E.,...@.      .g.F.
....P..d.....`..pG.....
19:55:39.121733 enp0s3 Out IP pop-os.51196 > 103.16.70.139.http: Flags [.], ack 1, win 64240, length 0
E..(..@.@..
...g.F....P....d...P.....
19:55:39.121984 enp0s3 Out IP pop-os.51196 > 103.16.70.139.http: Flags [P.], seq 1:422, ack 1, win 64240, length 421: HTTP: POST / HTTP/1.1
E....@.@..g
...g.F....P....d...P....i..POST / HTTP/1.1
Host: r3.o.lencr.org
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 85
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

0S0Q000M0K0  ..+.....H...+.~0..h..g.5.....XV..P  @.....b.....
19:55:39.122413 enp0s3 In  IP 103.16.70.139.http > pop-os.51196: Flags [.], ack 422, win 65535, length 0
E..(....@.      .g.F.
....P..d.....P.....
5 packets captured
7 packets received by filter
0 packets dropped by kernel
```

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

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

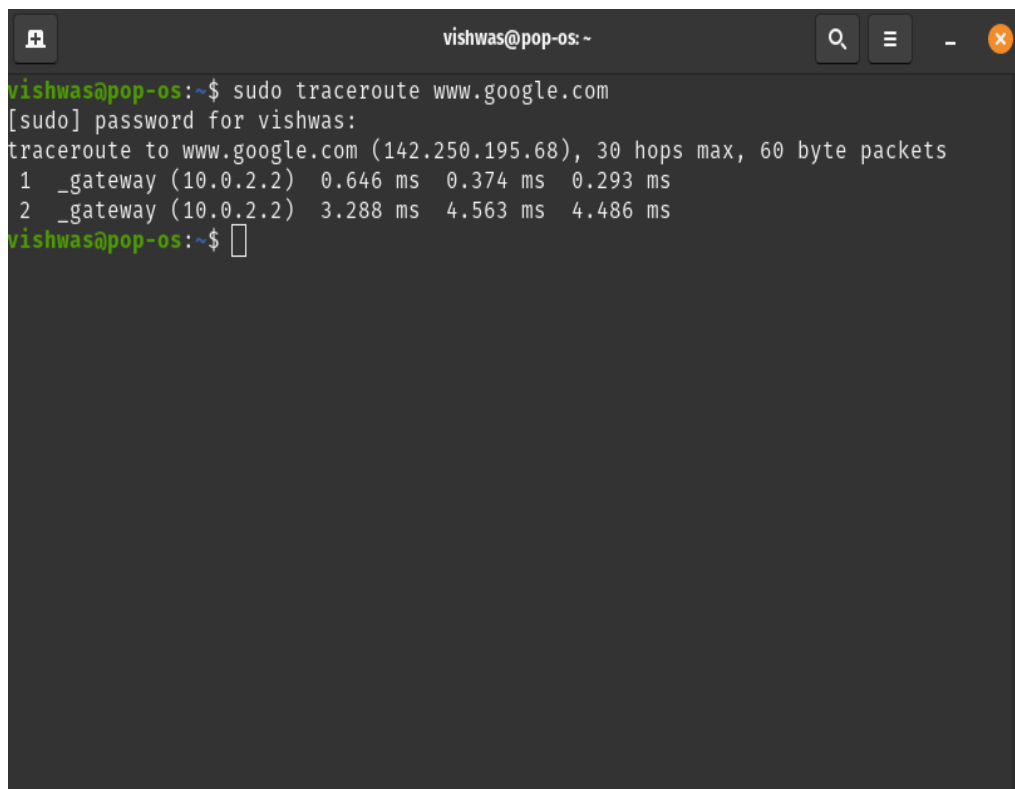
A terminal window titled 'vishwas@pop-os: ~' showing the execution of the 'sudo tcpdump' command. The output indicates that the data link type is LINUX_SLL2, it is listening on 'any', and it has captured 10 packets. The terminal text is as follows:

```
vishwas@pop-os:~$ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel
vishwas@pop-os:~$
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

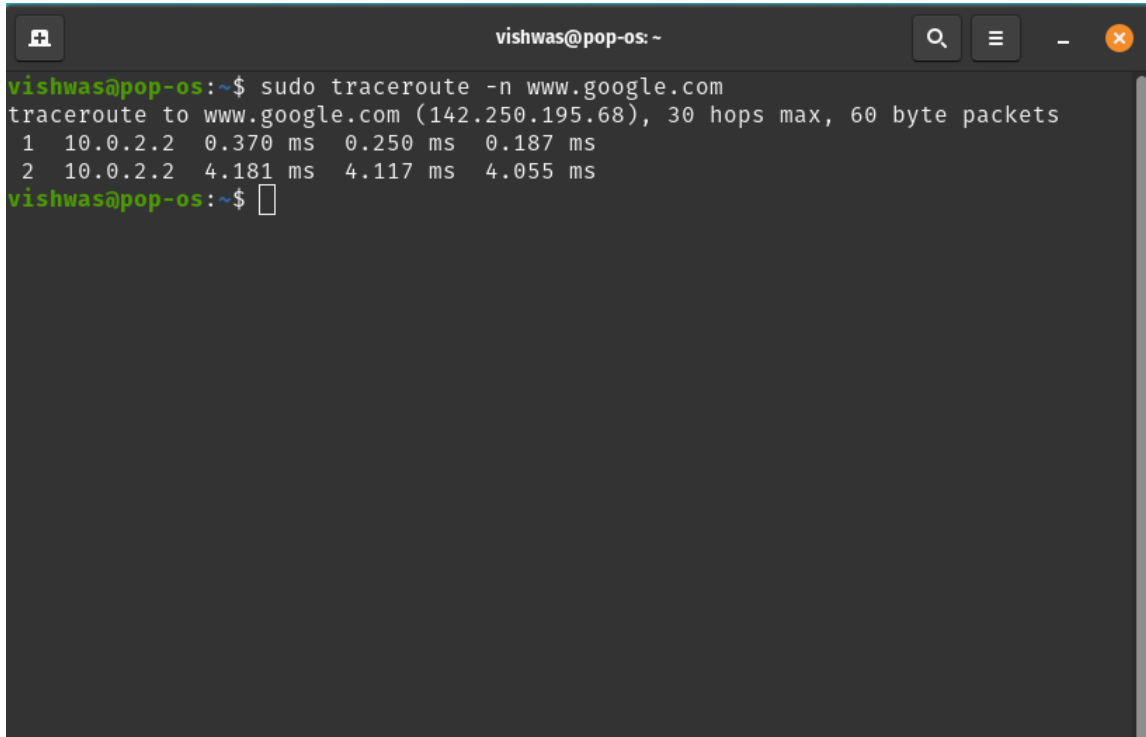
A terminal window titled 'vishwas@pop-os: ~' showing the execution of the 'sudo traceroute' command. The output shows the path to www.google.com (142.250.195.68) with 30 hops max and 60 byte packets. The first two hops are shown with their respective IP addresses and round-trip times. The terminal text is as follows:

```
vishwas@pop-os:~$ sudo traceroute www.google.com
[sudo] password for vishwas:
traceroute to www.google.com (142.250.195.68), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.646 ms  0.374 ms  0.293 ms
 2 _gateway (10.0.2.2)  3.288 ms  4.563 ms  4.486 ms
vishwas@pop-os:~$
```

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

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

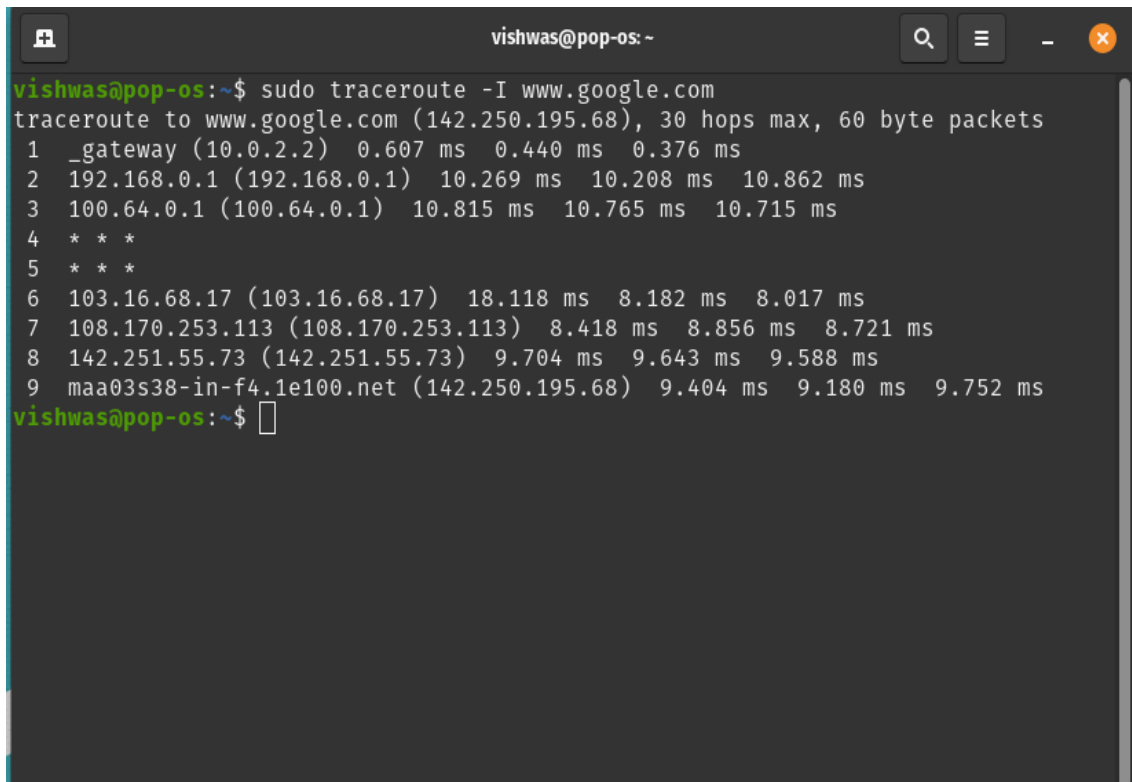
sudo traceroute -n www.google.com



```
vishwas@pop-os: ~  
vishwas@pop-os:~$ sudo traceroute -n www.google.com  
traceroute to www.google.com (142.250.195.68), 30 hops max, 60 byte packets  
 1  10.0.2.2  0.370 ms  0.250 ms  0.187 ms  
 2  10.0.2.2  4.181 ms  4.117 ms  4.055 ms  
vishwas@pop-os:~$
```

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

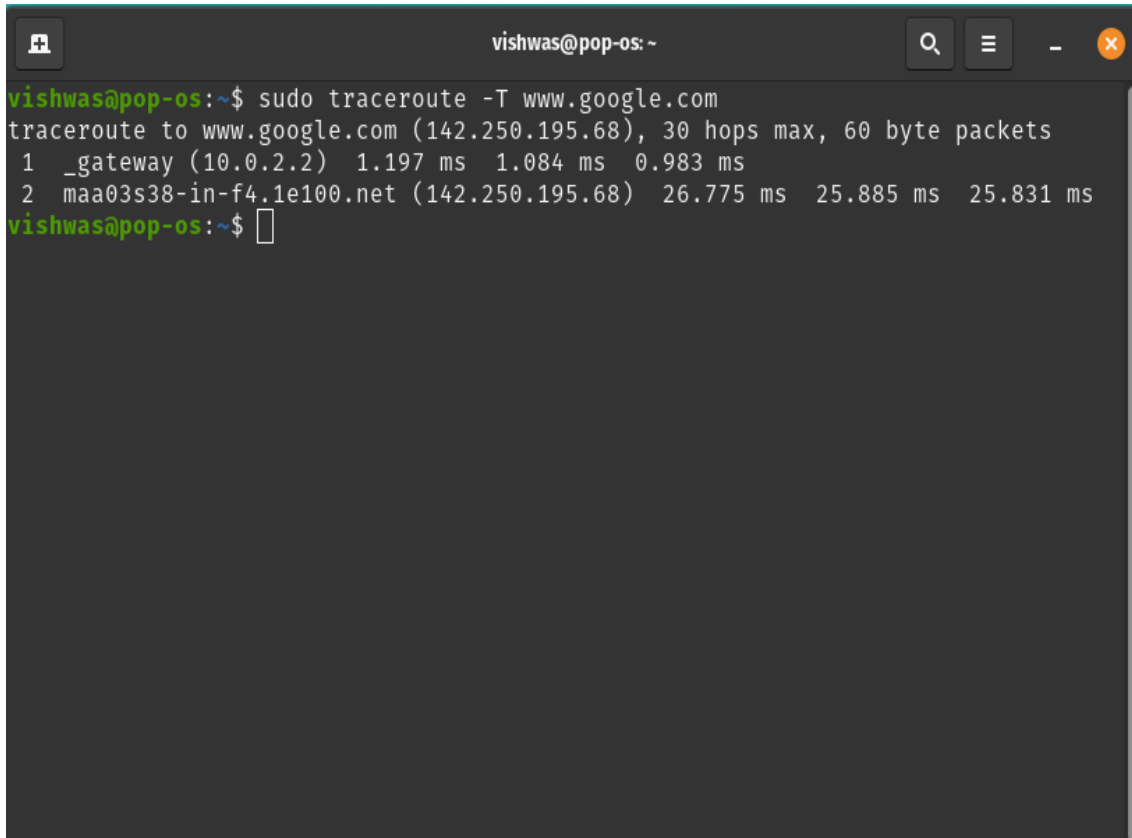
sudo traceroute -I www.google.com



```
vishwas@pop-os: ~  
vishwas@pop-os:~$ sudo traceroute -I www.google.com  
traceroute to www.google.com (142.250.195.68), 30 hops max, 60 byte packets  
 1  _gateway (10.0.2.2)  0.607 ms  0.440 ms  0.376 ms  
 2  192.168.0.1 (192.168.0.1)  10.269 ms  10.208 ms  10.862 ms  
 3  100.64.0.1 (100.64.0.1)  10.815 ms  10.765 ms  10.715 ms  
 4  * * *  
 5  * * *  
 6  103.16.68.17 (103.16.68.17)  18.118 ms  8.182 ms  8.017 ms  
 7  108.170.253.113 (108.170.253.113)  8.418 ms  8.856 ms  8.721 ms  
 8  142.251.55.73 (142.251.55.73)  9.704 ms  9.643 ms  9.588 ms  
 9  maa03s38-in-f4.1e100.net (142.250.195.68)  9.404 ms  9.180 ms  9.752 ms  
vishwas@pop-os:~$
```

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

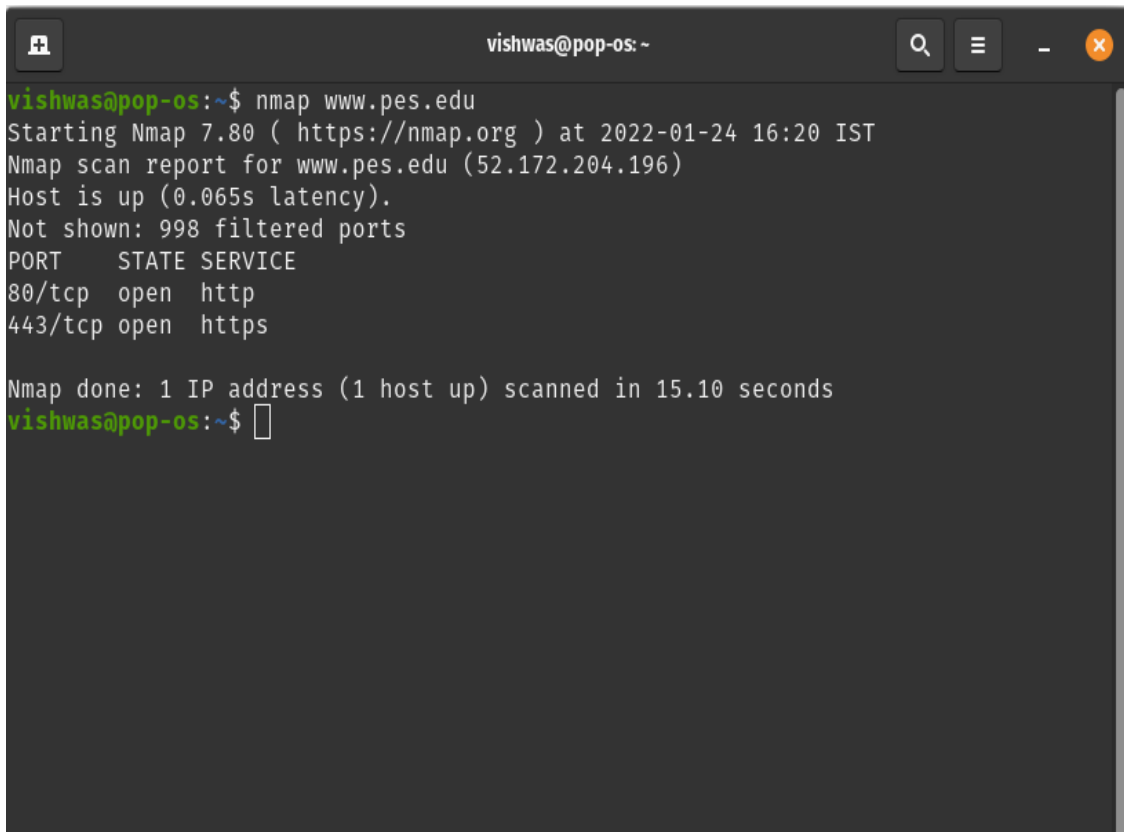
A terminal window titled 'vishwas@pop-os: ~' with search, menu, and window control buttons. The command 'sudo traceroute -T www.google.com' is entered. The output shows the traceroute path to www.google.com (142.250.195.68) with 30 hops max and 60 byte packets. The path consists of two hops: 1. _gateway (10.0.2.2) with 1.197 ms, 1.084 ms, and 0.983 ms; 2. maa03s38-in-f4.1e100.net (142.250.195.68) with 26.775 ms, 25.885 ms, and 25.831 ms.

```
vishwas@pop-os:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.195.68), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  1.197 ms  1.084 ms  0.983 ms
 2  maa03s38-in-f4.1e100.net (142.250.195.68)  26.775 ms  25.885 ms  25.831 ms
vishwas@pop-os:~$
```

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

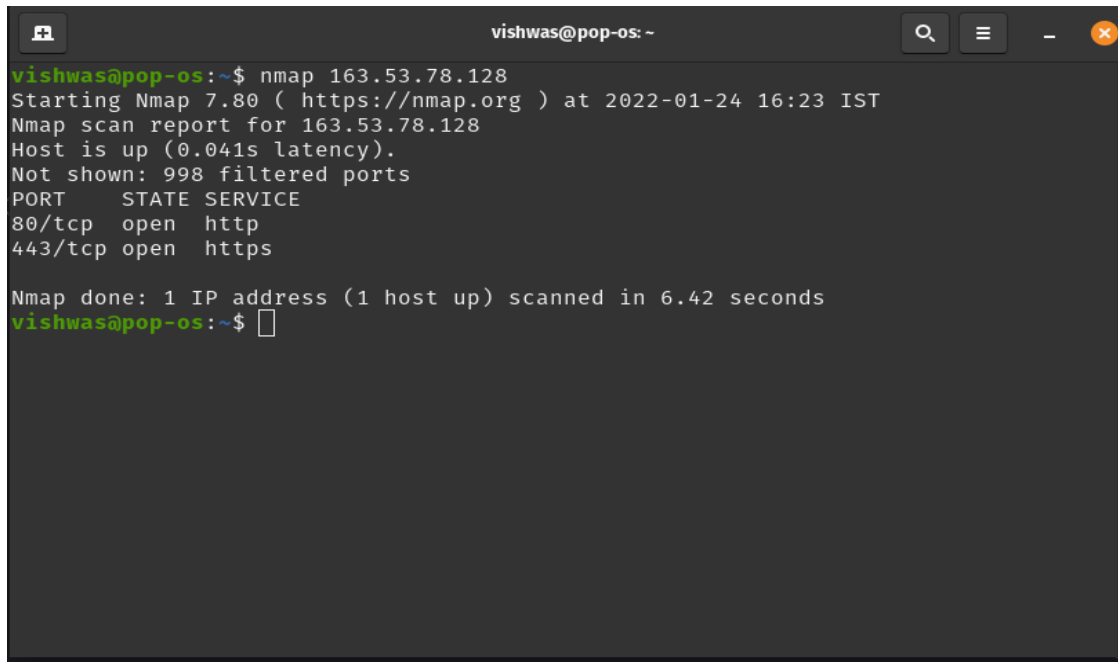
A terminal window titled 'vishwas@pop-os: ~' with search, menu, and window control buttons. The command 'nmap www.pes.edu' is entered. The output shows the Nmap scan report for www.pes.edu (52.172.204.196). The host is up with 0.065s latency. 998 filtered ports are not shown. Open ports are 80/tcp (http) and 443/tcp (https). The scan took 15.10 seconds.

```
vishwas@pop-os:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-24 16:20 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.065s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 15.10 seconds
vishwas@pop-os:~$
```

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

nmap 163.53.78.128

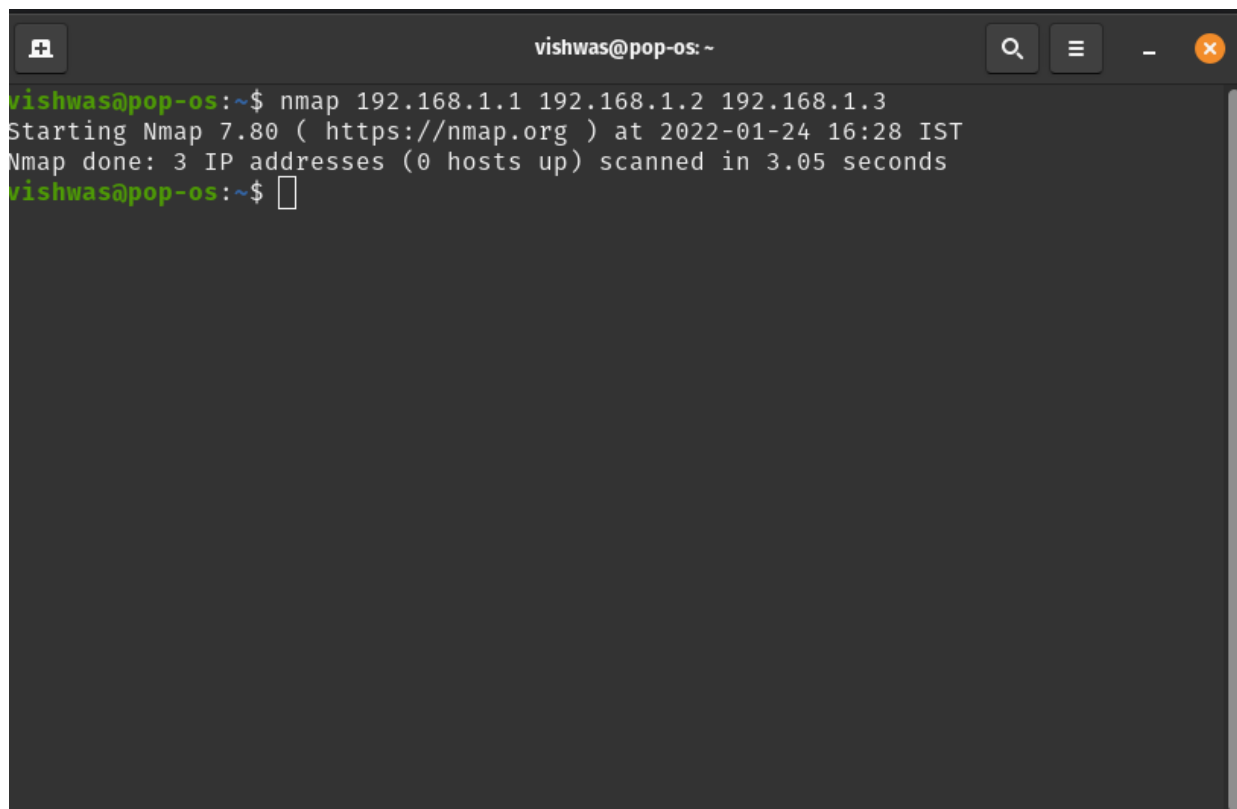
A terminal window titled 'vishwas@pop-os: ~' showing the execution of the nmap command. The output indicates that the host is up and lists open ports 80/tcp (http) and 443/tcp (https).

```
vishwas@pop-os:~$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-24 16:23 IST
Nmap scan report for 163.53.78.128
Host is up (0.041s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 6.42 seconds
vishwas@pop-os:~$
```

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

nmap 192.168.1.1 192.168.1.2 192.168.1.3

A terminal window titled 'vishwas@pop-os: ~' showing the execution of the nmap command with three IP addresses. The output shows that all three IP addresses were scanned, but no hosts were found to be up.

```
vishwas@pop-os:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-24 16:28 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.05 seconds
vishwas@pop-os:~$
```

Questions on above observations:

- 1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

Ans: 1.1. The version of the server is 1.1 as well.

- 2) When was the HTML file that you are retrieving last modified at the server?
Ans: Sun,23 Jan 2022 11:41:36 GMT

- 3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?
Ans: \$ ping -c <number of packets> <url>

- 4) How will you identify remote host apps and OS?
Ans: Simply scan the entire subnet.

Eg: \$ nmap -sP 10.0.4.*