PES UNIVERSITY

EC CAMPUS, BANGALORE

NAME SUHAN B REVANKAR

SRN PES2UG19CS412

WEEK 1

SUBJECT COMPUTER NETWORK LABORATORY

OBJECTIVE 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

```
suhan@suhan: ~
suhan@suhan:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:63:d9:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85895sec preferred_lft 85895sec
    inet6 fe80::fd91:7303:824c:c50f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

INTERFACE NAME	IP ADDRESS (IPV4/IPV6)	MAC ADDRESS
lo	IPV4: 127.0.0.1/8 IPV6: 1/128	00:00:00:00:00:00
enp0s3	IPV4:10.0.2.15/24 IPV6: fe80::fd91:7303:824c:c50f/64	08:00:27:63:d9:f5

Step 2: To assign an IP address to an interface

```
suhan@suhan:~$ sudo ifconfig enp0s3 10.0.7.11 netmask 255.255.255.0
[sudo] password for suhan:
```

Step 3: To activate / deactivate a network interface

```
suhan@suhan:~$ sudo ifconfig lo up
suhan@suhan:~$ sudo ifconfig enp0s3 up
```

Step 4: To show the current neighbor table in kernel

```
suhan@suhan:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
```

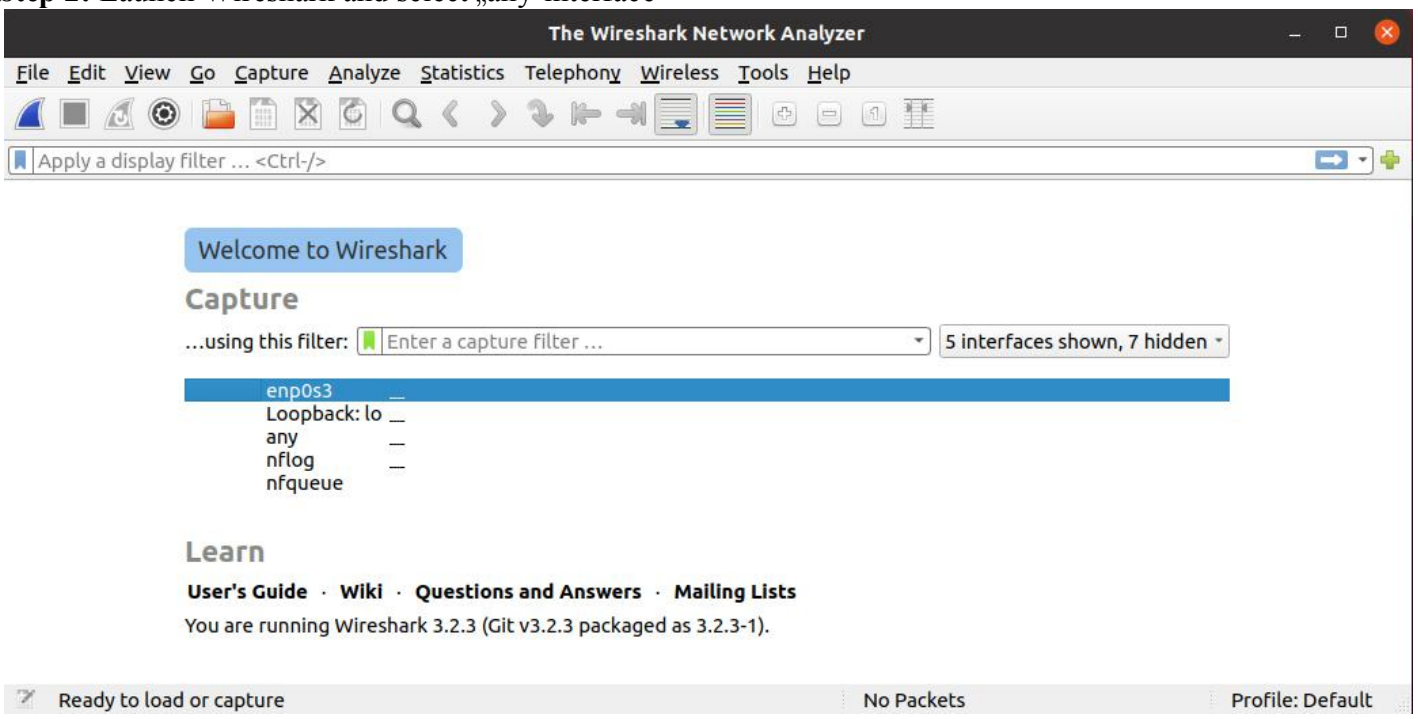
Shows neighbor objects as REACHABLE

TASK 2: PING PDU (PACKET DATA UNITS OR PACKETS) CAPTURE

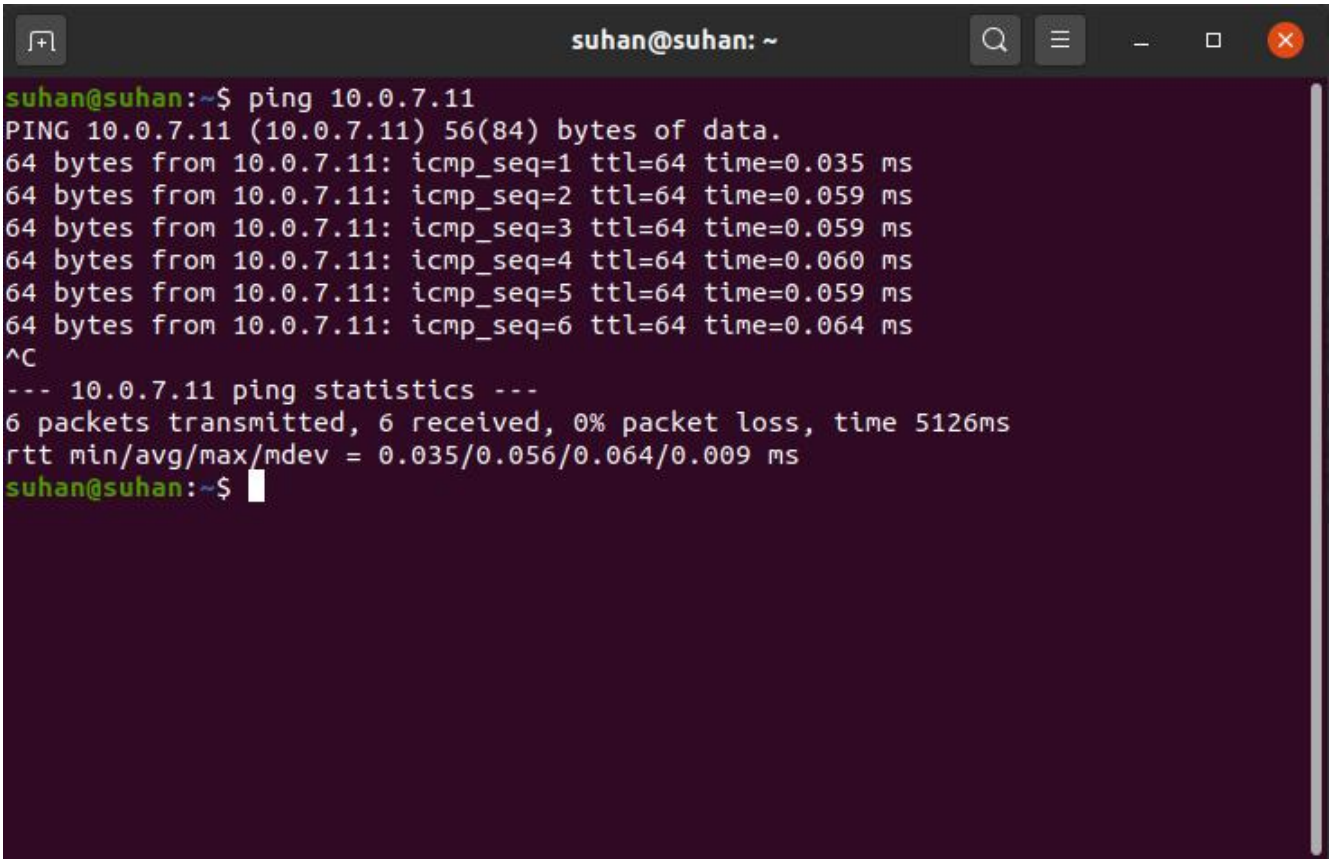
Step 1: Assign an IP address to the system (Host).

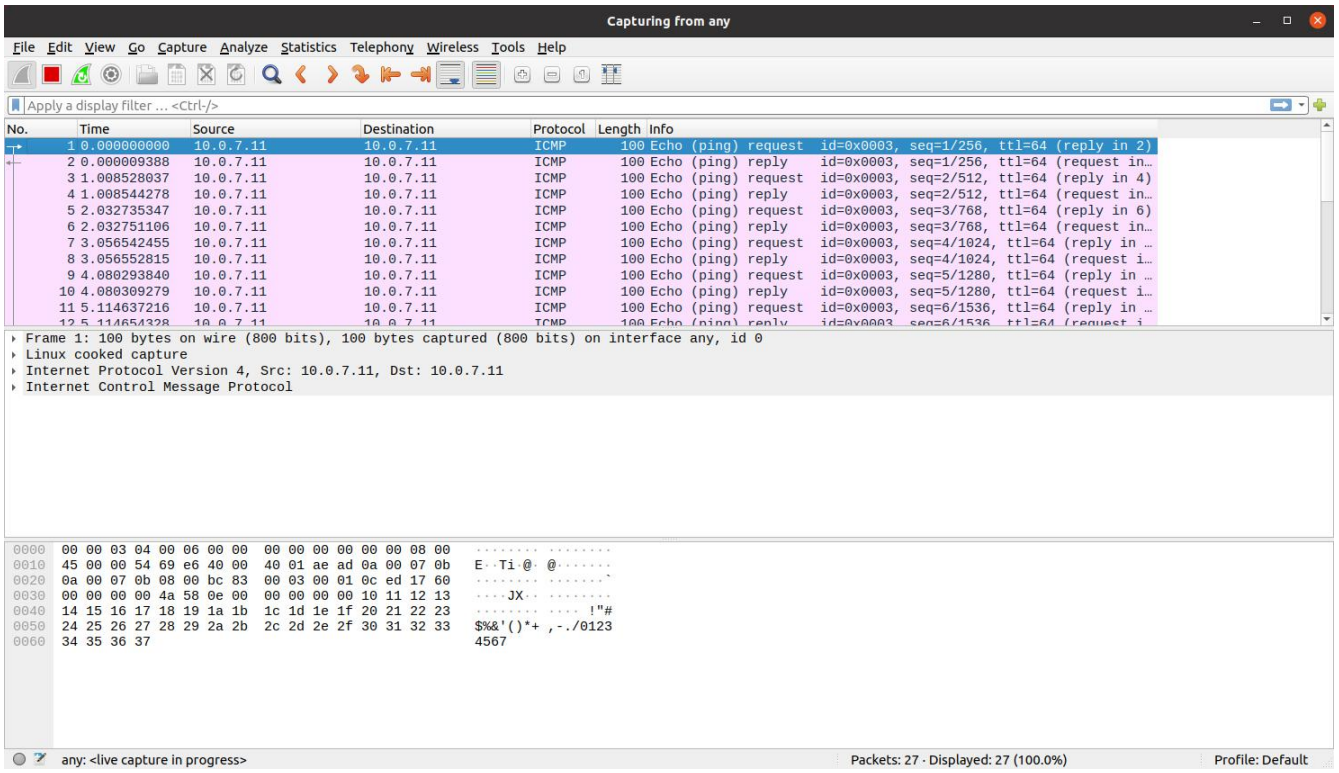
Note: IP address of your system should be 10.0. your_section. your_sno.

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: Analyse the following in Terminal

- **TTL** - 64
- **PROTOCOL USED BY PING** - ICMP
- **TIME** - 5126ms

Step 5: Analyse the following in Wireshark

Details	**First Echo Request**	**First Echo Reply**
Frame Number	1	2
Source IP address	10.0.7.11	10.0.7.11
Destination IP address	10.0.7.11	10.0.7.11
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
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	64

TASK 3: HTTP PDU CAPTURE

Using Wireshark"s Filter feature

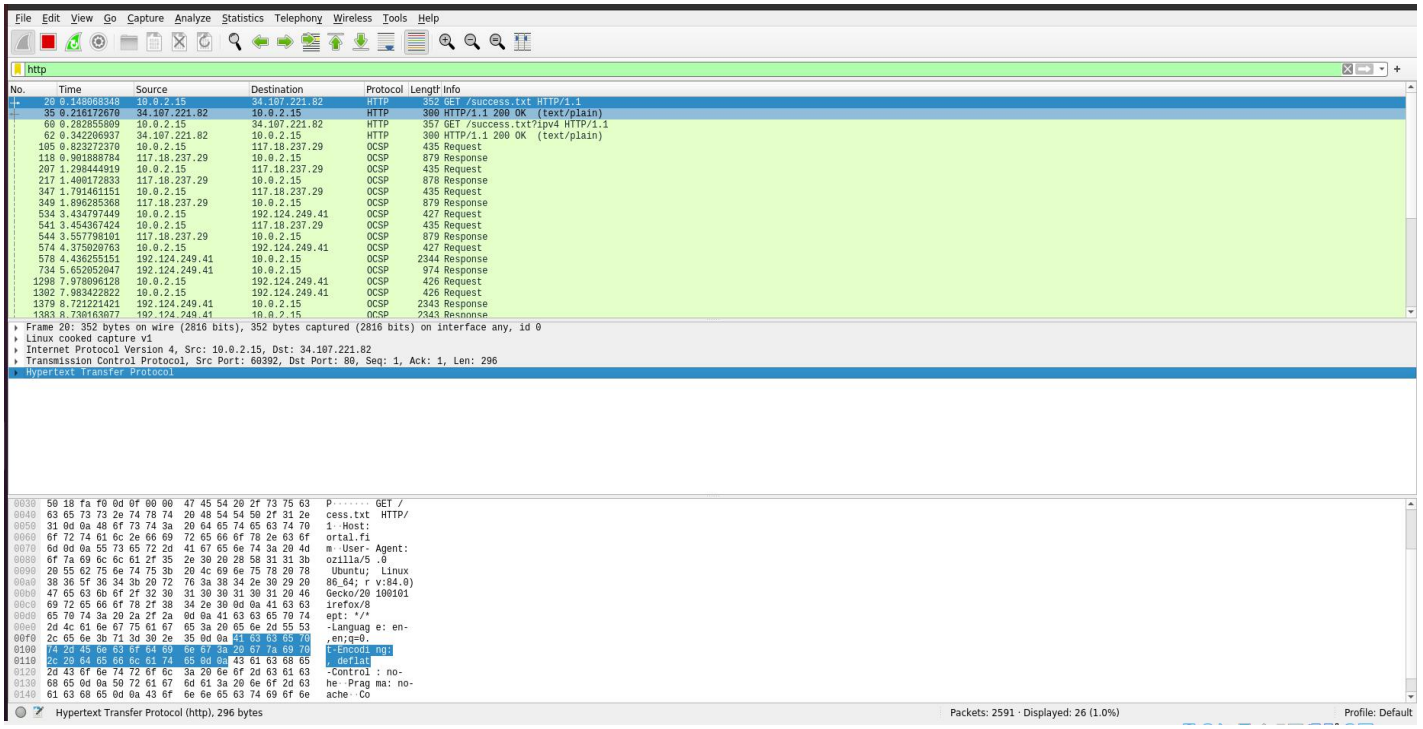
Step 1: Launch Wireshark and select „any' interface. On the Filter toolbar, type-in „http" and press enter

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:

Details	**First Echo Request**	**First Echo Reply**
Frame Number	20	35
Source IP address	10.0.2.15	34.107.221.82
Destination IP address	34.107.221.82	10.0.2.15
ICMP Type Value	IPv4	IPv4
ICMP Code Value	0x0800	0x0800
Source Ethernet Address	PcsCompu_9d:cc:ee (08:00:27:9d:cc:ee)	RealtekU_12:35:02 (52:54:00:12:35:02)
Destination Ethernet Address	RealtekU_12:35:02 (52:54:00:12:35:02)	PcsCompu_9d:cc:ee (08:00:27:9d:cc:ee)
Internet Protocol Version	4	4
Time To Live (TTL) Value	64	64



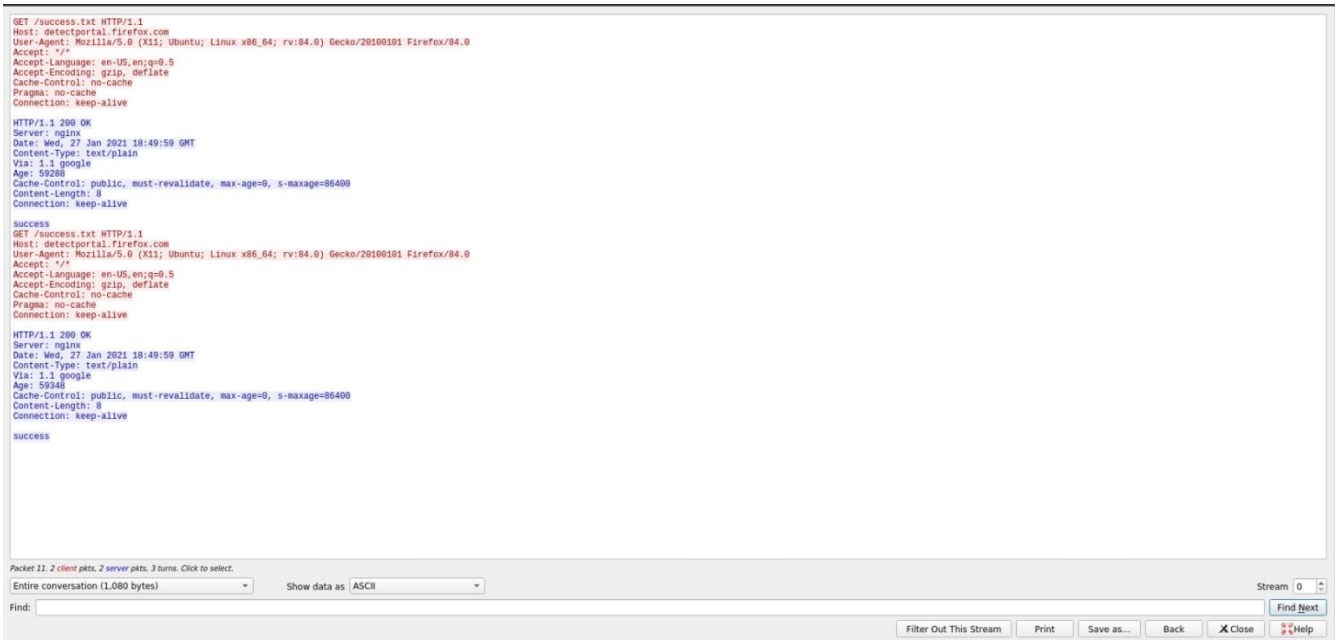
Step 4: Analyze the HTTP request and response and complete the table below.

HTTP Request		**HTTP Response**	
Get	GET / HTTP / 1.1	**Server**	ECS(tir/CCD5)\r\n
Host	connectivity-check.ubuntu.com	**Content-Type**	text/html\r\n
User-Agent	Mozilla/5.0(ubuntu)	**Date**	Thur 28 Jan 2021
Accept-Language	en-US	**Location**	
Accept-Encoding	gzip,deflate	**Content-Length**	471\r\n
Connection	keep-alive\r\n	**Connection**	keep-alive\r\n

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

```
suhan@suhan: ~
suhan@suhan:~$ sudo tcpdump -D
[sudo] password for suhan:
1.enp0s3 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
```

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.

OBSERVATION

Step 3: Understand the output format.

Capture all packets in any interface by running this command:

```
suhan@suhan: ~
suhan@suhan:~$ ping -c 6 google.com & sudo tcpdump -i any
[1] 4567
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
17:56:18.199821 IP localhost.34640 > localhost.domain: 21681+ [1au] A? api.snapcraft.io. (45)
17:56:18.199928 IP localhost.34640 > localhost.domain: 51882+ [1au] AAAA? api.snapcraft.io. (45)
17:56:18.200803 IP localhost.47371 > localhost.domain: 60419+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
17:56:18.200921 IP localhost.domain > localhost.47371: 60419 1/0/1 PTR localhost. (75)
17:56:19.547429 IP localhost.55610 > localhost.domain: 19167+ [1au] A? google.com. (39)
17:56:19.547545 IP localhost.55610 > localhost.domain: 2518+ [1au] AAAA? google.com. (39)
ping: google.com: Temporary failure in name resolution
17:56:31.023903 IP localhost.34486 > localhost.domain: 15186+ [1au] A? api.snapcraft.io. (45)
17:56:31.023931 IP localhost.34486 > localhost.domain: 57386+ [1au] AAAA? api.snapcraft.io. (45)
17:56:36.029524 IP localhost.34486 > localhost.domain: 15186+ [1au] A? api.snapcraft.io. (45)
17:56:36.029586 IP localhost.34486 > localhost.domain: 57386+ [1au] AAAA? api.snapcraft.io. (45)
17:57:02.394273 IP localhost.38593 > localhost.domain: 42287+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:02.394304 IP localhost.38593 > localhost.domain: 808+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:07.423534 IP localhost.38593 > localhost.domain: 42287+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:07.423566 IP localhost.38593 > localhost.domain: 808+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:12.429037 IP localhost.44504 > localhost.domain: 24719+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:12.429067 IP localhost.44504 > localhost.domain: 60551+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:17.449424 IP localhost.44504 > localhost.domain: 24719+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:17.449458 IP localhost.44504 > localhost.domain: 60551+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:22.455735 IP localhost.47252 > localhost.domain: 30024+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:22.455757 IP localhost.47252 > localhost.domain: 6727+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:27.472789 IP localhost.47252 > localhost.domain: 30024+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:27.472863 IP localhost.47252 > localhost.domain: 6727+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:32.503810 IP localhost.42559 > localhost.domain: 61952+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:32.503840 IP localhost.42559 > localhost.domain: 34586+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:37.530071 IP localhost.42559 > localhost.domain: 61952+ [1au] A? safebrowsing.googleapis.com. (56)
17:57:37.530155 IP localhost.42559 > localhost.domain: 34586+ [1au] AAAA? safebrowsing.googleapis.com. (56)
17:57:40.191836 IP localhost.41524 > localhost.domain: 43713+ [1au] A? incoming.telemetry.mozilla.org. (59)
17:57:40.191859 IP localhost.41524 > localhost.domain: 19400+ [1au] AAAA? incoming.telemetry.mozilla.org. (59)
17:57:45.196538 IP localhost.41524 > localhost.domain: 43713+ [1au] A? incoming.telemetry.mozilla.org. (59)
17:57:45.196561 IP localhost.41524 > localhost.domain: 19400+ [1au] AAAA? incoming.telemetry.mozilla.org. (59)
17:57:50.200628 IP localhost.36656 > localhost.domain: 14927+ [1au] A? incoming.telemetry.mozilla.org. (59)
17:57:50.201376 IP localhost.36656 > localhost.domain: 35398+ [1au] AAAA? incoming.telemetry.mozilla.org. (59)
17:57:55.205430 IP localhost.36656 > localhost.domain: 14927+ [1au] A? incoming.telemetry.mozilla.org. (59)
17:57:55.205456 IP localhost.36656 > localhost.domain: 35398+ [1au] AAAA? incoming.telemetry.mozilla.org. (59)
17:57:59.716093 IP localhost.38386 > localhost.domain: 21175+ [1au] A? normandy.cdn.mozilla.net. (53)
17:57:59.716223 IP localhost.38386 > localhost.domain: 46271+ [1au] AAAA? normandy.cdn.mozilla.net. (53)
17:58:00.213058 IP localhost.34150 > localhost.domain: 1279+ [1au] A? incoming.telemetry.mozilla.org. (59)
17:58:00.213916 IP localhost.34150 > localhost.domain: 16624+ [1au] AAAA? incoming.telemetry.mozilla.org. (59)
```

```
suhan@suhan: ~
suhan@suhan:~$ ping -c 6 google.com & sudo tcpdump -i any
[1] 2578
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
PING google.com (142.250.183.78) 56(84) bytes of data.
18:53:33.677714 IP 192.168.43.222.domain > suhan.59384: 41588 1/0/0 A 142.250.183.78 (44)
18:53:33.678332 IP 192.168.43.222.domain > suhan.48970: 13192 1/0/0 AAAA 2404:6800:4007:813::200e (56)
18:53:33.678373 IP localhost.domain > localhost.44348: 5794 1/0/1 A 142.250.183.78 (55)
18:53:33.678556 IP localhost.domain > localhost.44348: 682 1/0/1 AAAA 2404:6800:4007:813::200e (67)
18:53:33.678732 IP localhost.40606 > localhost.domain: 5047+ [1au] PTR? 15.2.0.10.in-addr.arpa. (51)
18:53:33.999442 IP localhost.58782 > localhost.domain: 31763+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
18:53:34.722908 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 1, length 64
18:53:34.723591 IP localhost.41099 > localhost.domain: 20142+ [1au] PTR? 78.183.250.142.in-addr.arpa. (56)
18:53:34.723634 IP localhost.54685 > localhost.domain: 55864+ [1au] PTR? 78.183.250.142.in-addr.arpa. (56)
18:53:34.724032 IP suhan.54124 > 192.168.43.222.domain: 36995+ PTR? 78.183.250.142.in-addr.arpa. (45)
18:53:34.769215 IP 192.168.43.222.domain > suhan.54124: 36995 1/0/0 PTR bom12s12-in-f14.1e100.net. (84)
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=1 ttl=117 time=1044 ms
18:53:34.770625 IP suhan > bom12s12-in-f14.1e100.net: ICMP echo request, id 1, seq 2, length 64
18:53:34.839437 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 2, length 64
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=2 ttl=117 time=68.8 ms
18:53:35.771690 IP suhan > bom12s12-in-f14.1e100.net: ICMP echo request, id 1, seq 3, length 64
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=3 ttl=117 time=73.4 ms
18:53:35.845021 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 3, length 64
18:53:36.775601 IP suhan > bom12s12-in-f14.1e100.net: ICMP echo request, id 1, seq 4, length 64
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=4 ttl=117 time=88.1 ms
18:53:36.863657 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 4, length 64
18:53:37.777525 IP suhan > bom12s12-in-f14.1e100.net: ICMP echo request, id 1, seq 5, length 64
18:53:37.856124 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 5, length 64
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=5 ttl=117 time=78.7 ms
18:53:38.780258 IP suhan > bom12s12-in-f14.1e100.net: ICMP echo request, id 1, seq 6, length 64
18:53:38.847489 IP bom12s12-in-f14.1e100.net > suhan: ICMP echo reply, id 1, seq 6, length 64
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=6 ttl=117 time=67.3 ms

--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5101ms
rtt min/avg/max/mdev = 67.267/236.644/1043.628/360.959 ms
18:53:39.813783 ARP, Request who-has _gateway tell suhan, length 28
18:53:39.814072 ARP, Reply _gateway is-at 52:54:00:12:35:02 (oui Unknown), length 46
18:53:39.814260 IP localhost.39615 > localhost.domain: 42422+ [1au] PTR? 2.2.0.10.in-addr.arpa. (50)
18:53:39.814930 IP suhan.45567 > 192.168.43.222.domain: 36173+ PTR? 2.2.0.10.in-addr.arpa. (39)
18:53:39.823072 IP 192.168.43.222.domain > suhan.45567: 36173 NXDomain 0/0/0 (39)
```

Listen, report the list of link-layer types, report the list of time stamp types, or report the results of compiling a filter expression on interface.

Step 4: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by : sudo tcpdump -I any -c5 icmp

```
suhan@suhan:~$ sudo tcpdump -i any -c5 icmp -v
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

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

```
suhan@suhan:~$ 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
18:57:53.399834 IP 10.0.2.15.54520 > 35.224.170.84.80: Flags [S], seq 3531531667, win 64240, options [mss 1460,sackOK,TS val 2212503152 ecr 0,nop,wscale 7], length 0
18:57:54.021282 IP 35.224.170.84.80 > 10.0.2.15.54520: Flags [S.], seq 16320001, ack 3531531668, win 65535, options [mss 1460], length 0
18:57:54.021383 IP 10.0.2.15.54520 > 35.224.170.84.80: Flags [.], ack 1, win 64240, length 0
18:57:54.021881 IP 10.0.2.15.54520 > 35.224.170.84.80: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HTTP: GET / HTTP/1.1
18:57:54.022599 IP 35.224.170.84.80 > 10.0.2.15.54520: Flags [.], ack 88, win 65535, length 0
18:57:54.369756 IP 35.224.170.84.80 > 10.0.2.15.54520: Flags [P.], seq 1:149, ack 88, win 65535, length 148: HTTP: HTTP/1.1 204 No Content
18:57:54.369806 IP 10.0.2.15.54520 > 35.224.170.84.80: Flags [.], ack 149, win 64092, length 0
18:57:54.370139 IP 10.0.2.15.54520 > 35.224.170.84.80: Flags [F.], seq 88, ack 149, win 64092, length 0
18:57:54.370671 IP 35.224.170.84.80 > 10.0.2.15.54520: Flags [.], ack 89, win 65535, length 0
18:57:54.373183 IP 35.224.170.84.80 > 10.0.2.15.54520: Flags [F.], seq 149, ack 89, win 65535, length 0
10 packets captured
11 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

```
suhan@suhan:~$ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel
```

TASK 5: PERFORM TRACEROUTE CHECKS

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

```
suhan@suhan: ~
suhan@suhan:~$ sudo traceroute www.google.com
[sudo] password for suhan:
traceroute to www.google.com (142.250.76.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.728 ms  0.692 ms  0.662 ms
 2  _gateway (10.0.2.2)  225.011 ms  220.290 ms  225.190 ms
```

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

The destination address is 172.217.26.164 and there were 30 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

```
suhan@suhan:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.76.164), 30 hops max, 60 byte packets
 1  10.0.2.2  0.356 ms  0.281 ms  1.304 ms
 2  10.0.2.2  100.907 ms  147.696 ms  144.492 ms
```

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

sudo traceroute -I www.google.com

```
suhan@suhan:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.76.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.650 ms  0.612 ms  0.603 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * 72.14.208.234 (72.14.208.234)  109.087 ms  109.908 ms
 7  74.125.242.129 (74.125.242.129)  109.908 ms  109.818 ms  109.753 ms
 8  74.125.242.139 (74.125.242.139)  110.695 ms  110.687 ms  110.671 ms
 9  72.14.236.174 (72.14.236.174)  138.760 ms  138.762 ms  139.336 ms
10  108.170.248.161 (108.170.248.161)  137.539 ms  148.801 ms  149.461 ms
11  216.239.46.137 (216.239.46.137)  149.459 ms  154.727 ms  155.891 ms
12  bom12s09-in-f4.1e100.net (142.250.76.164)  96.194 ms  110.675 ms  109.969 ms
```

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

```
suhan@suhan:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.76.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  6.202 ms  6.161 ms  6.149 ms
 2  bom12s09-in-f4.1e100.net (142.250.76.164)  265.031 ms  257.656 ms  313.241 ms
```

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

```
suhan@suhan: ~
suhan@suhan:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-01 18:49 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.11s latency).
```

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

nmap 163.53.78.128

```
suhan@suhan:~$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-01 18:50 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds
```

Step 3: Scan multiple IP address or subnet

(IPv4) nmap 192.168.1.1 192.168.1.2

192.168.1.3

```
suhan@suhan:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-01 18:51 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.07 seconds
suhan@suhan:~$
```