Week #1

Name: Akash G Gaonkar **SECTION:A**

SRN: PES2UG22CS043

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 as a single PDF file. (The PDF must contain: Lab Number and Title, SRN and Name of the student, Section)
- 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, thenyour 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:

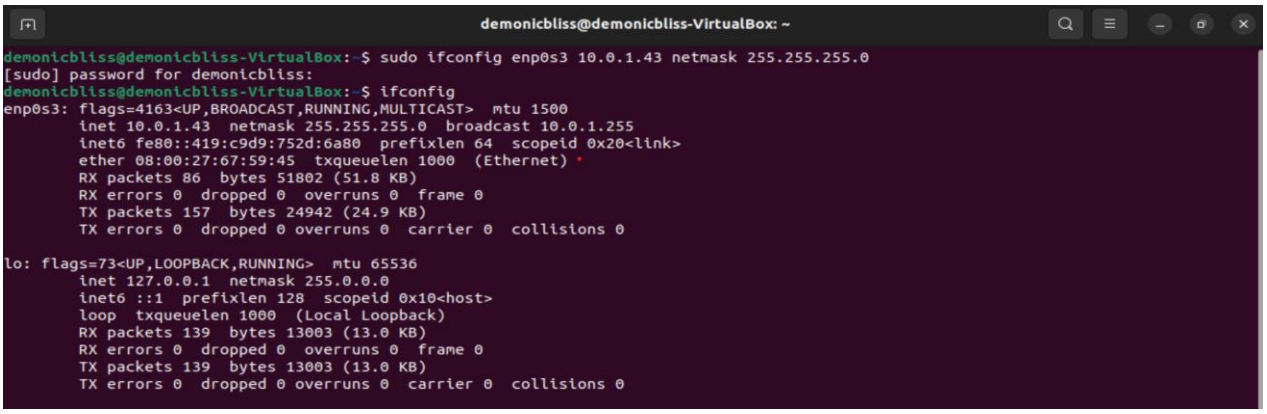
ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address	
enp0s3	10.0.2.12	08:00:27:67:59:45	
lo	127.0.0.1	NA	

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



IP address has been changed .

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

sudo ifconfig interface_name down

sudo ifconfig interface_name up

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo ifconfig enp0s3 down
demonicbliss@demonicbliss-VirtualBox:~$ ifconfig
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 177  bytes 16358 (16.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 177  bytes 16358 (16.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

demonicbliss@demonicbliss-VirtualBox:~$ sudo ifconfig enp0s3 up
demonicbliss@demonicbliss-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::419:c9d9:752d:6a80  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:67:59:45  txqueuelen 1000  (Ethernet)
        RX packets 95  bytes 53534 (53.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 206  bytes 30098 (30.0 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 177  bytes 16358 (16.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 177  bytes 16358 (16.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

IP address has been restored to the original value

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

ip neigh

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

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.

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo ifconfig enp0s3 10.0.1.43 netmask 255.255.255.0
demonicbliss@demonicbliss-VirtualBox:~$ sudo wireshark
 ** (wireshark:2202) 12:52:44.957478 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
 ** (wireshark:2202) 12:52:55.114693 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2202) 12:52:55.231949 [Capture MESSAGE] -- Capture started
 ** (wireshark:2202) 12:52:55.232103 [Capture MESSAGE] -- File: "/tmp/wireshark_anyIVNWH2.pcapng"
 ** (wireshark:2202) 12:52:57.956834 [Capture MESSAGE] -- Capture Stop ...
 ** (wireshark:2202) 12:52:58.341061 [Capture MESSAGE] -- Capture stopped.
 ** (wireshark:2202) 12:53:05.536847 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2202) 12:53:05.653603 [Capture MESSAGE] -- Capture started
 ** (wireshark:2202) 12:53:05.653656 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3155NH2.pcapng"
 ** (wireshark:2202) 12:53:07.461578 [Capture MESSAGE] -- Capture Stop ...
 ** (wireshark:2202) 12:53:07.534615 [Capture MESSAGE] -- Capture stopped.
 ** (wireshark:2202) 12:53:11.591652 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:2202) 12:53:11.690339 [Capture MESSAGE] -- Capture started
 ** (wireshark:2202) 12:53:11.690696 [Capture MESSAGE] -- File: "/tmp/wireshark_any7LYJH2.pcapng"
```

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
- Protocol used by ping
- Time

```
demonicbliss@demonicbliss-VirtualBox:~$ ping 10.0.1.43
PING 10.0.1.43 (10.0.1.43) 56(84) bytes of data.
64 bytes from 10.0.1.43: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 10.0.1.43: icmp_seq=2 ttl=64 time=0.092 ms
64 bytes from 10.0.1.43: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.0.1.43: icmp_seq=4 ttl=64 time=0.103 ms
64 bytes from 10.0.1.43: icmp_seq=5 ttl=64 time=0.067 ms
64 bytes from 10.0.1.43: icmp_seq=6 ttl=64 time=0.066 ms
64 bytes from 10.0.1.43: icmp_seq=7 ttl=64 time=0.060 ms
64 bytes from 10.0.1.43: icmp_seq=8 ttl=64 time=0.093 ms
64 bytes from 10.0.1.43: icmp_seq=9 ttl=64 time=0.082 ms
64 bytes from 10.0.1.43: icmp_seq=10 ttl=64 time=0.075 ms
64 bytes from 10.0.1.43: icmp_seq=11 ttl=64 time=0.082 ms
64 bytes from 10.0.1.43: icmp_seq=12 ttl=64 time=0.079 ms
64 bytes from 10.0.1.43: icmp_seq=13 ttl=64 time=0.088 ms
64 bytes from 10.0.1.43: icmp_seq=14 ttl=64 time=0.065 ms
64 bytes from 10.0.1.43: icmp_seq=15 ttl=64 time=0.083 ms
64 bytes from 10.0.1.43: icmp_seq=16 ttl=64 time=0.097 ms
64 bytes from 10.0.1.43: icmp_seq=17 ttl=64 time=0.076 ms
64 bytes from 10.0.1.43: icmp_seq=18 ttl=64 time=0.074 ms
64 bytes from 10.0.1.43: icmp_seq=19 ttl=64 time=0.068 ms
64 bytes from 10.0.1.43: icmp_seq=20 ttl=64 time=0.107 ms
64 bytes from 10.0.1.43: icmp_seq=21 ttl=64 time=0.092 ms
64 bytes from 10.0.1.43: icmp_seq=22 ttl=64 time=0.081 ms
64 bytes from 10.0.1.43: icmp_seq=23 ttl=64 time=0.088 ms
64 bytes from 10.0.1.43: icmp_seq=24 ttl=64 time=0.076 ms
64 bytes from 10.0.1.43: icmp_seq=25 ttl=64 time=0.091 ms
64 bytes from 10.0.1.43: icmp_seq=26 ttl=64 time=0.080 ms
64 bytes from 10.0.1.43: icmp_seq=27 ttl=64 time=0.057 ms
64 bytes from 10.0.1.43: icmp_seq=28 ttl=64 time=0.394 ms
64 bytes from 10.0.1.43: icmp_seq=29 ttl=64 time=0.066 ms
64 bytes from 10.0.1.43: icmp_seq=30 ttl=64 time=0.066 ms
```

ping command has been made to my own system

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	21	22
Source IP address	10.0.1.43	10.0.1.43
Destination IP address	10.0.1.43	10.0.1.43
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	38	39
Source Port	55764	443
Destination Port	443	55764
Source IP address	192.168.1.12	103.243.33.5
Destination IP address	103.243.33.5	192.168.1.12
Source Ethernet Address	08:00:27:67:59:45	b4:3d:08:59:9f:98
Destination Ethernet Address	NA	NA

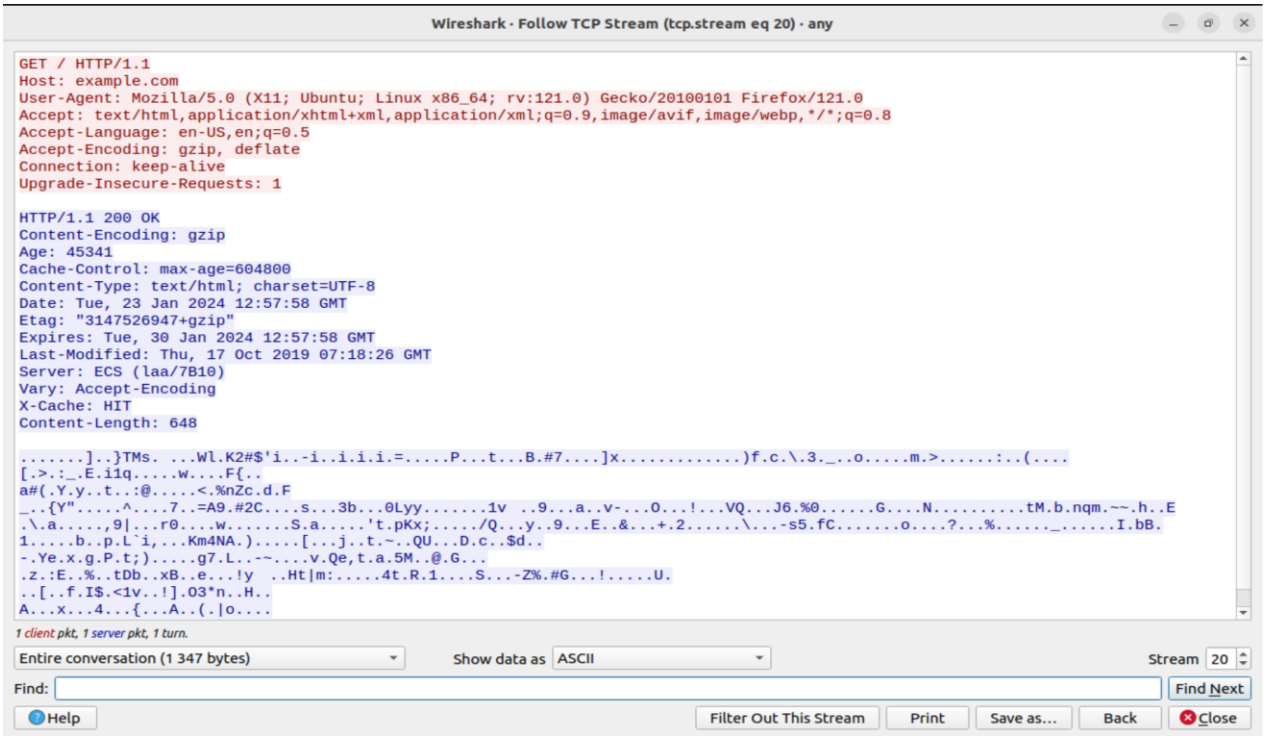
Step 4: Analyze the HTTP request and response and complete the table below.

HTTP Request		HTTP Response	
Get	/http/1.1	Server	HTTP/1.1 200 OK
Host	example.com	Content-Type	Text/html
User-Agent	Mozilla/5.0 (K11) Ubuntu Linux 408 64, rv 121.0) Gecko/20100101 Firefox/121.0	Date	Tue,23 Jan2924 12:57:58 GMT
Accept-Language	en-US	Location	NA
Accept-Encoding	Gzip	Content-Length	648
Connection	Keep-alive	Connection	NA

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

```
root@demonicbliss-VirtualBox:~# sudo tcpdump -d
Warning: assuming Ethernet
(000) ret      #262144
```

Tcp dump command assumes Ethernet connection since we are operating in a virtual machine

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.

```
root@demonicbliss-VirtualBox:~# sudo tcpdump -i any
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
14:01:42.173890 enp0s3 B   ARP, Request who-has _gateway tell 192.168.1.3, length 46
14:01:42.277113 lo    In  IP localhost.42153 > localhost.domain: 18522+ [1au] PTR? 1.1.168.192.in-addr.arpa. (53)
14:01:42.278278 enp0s3 Out IP demonicbliss-VirtualBox.38035 > _gateway.domain: 49762+ [1au] PTR? 1.1.168.192.in-addr.arpa. (53)
14:01:42.285583 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.38035: 49762 NXDomain* 0/1/1 (108)
14:01:42.286103 enp0s3 Out IP demonicbliss-VirtualBox.38035 > _gateway.domain: 49762+ PTR? 1.1.168.192.in-addr.arpa. (42)
14:01:42.292596 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.38035: 49762 NXDomain* 0/1/0 (97)
14:01:42.293587 lo    In  IP localhost.domain > localhost.42153: 18522*$ 1/0/1 PTR _gateway. (75)
14:01:42.294219 lo    In  IP localhost.33685 > localhost.domain: 59049+ [1au] PTR? 3.1.168.192.in-addr.arpa. (53)
14:01:42.294855 enp0s3 Out IP demonicbliss-VirtualBox.39388 > _gateway.domain: 17082+ [1au] PTR? 3.1.168.192.in-addr.arpa. (53)
14:01:42.300939 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.39388: 17082 NXDomain* 0/1/1 (108)
14:01:42.301457 enp0s3 Out IP demonicbliss-VirtualBox.39388 > _gateway.domain: 17082+ PTR? 3.1.168.192.in-addr.arpa. (42)
14:01:42.310818 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.39388: 17082 NXDomain* 0/1/0 (97)
14:01:42.311581 lo    In  IP localhost.domain > localhost.33685: 59049 NXDomain 0/1/1 (88)
14:01:42.395580 lo    In  IP localhost.53142 > localhost.domain: 15037+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
14:01:42.396001 lo    In  IP localhost.domain > localhost.53142: 15037*$ 1/0/1 PTR localhost. (75)
14:01:42.398482 lo    In  IP localhost.35638 > localhost.domain: 40232+ [1au] PTR? 12.1.168.192.in-addr.arpa. (54)
14:01:42.399141 enp0s3 Out IP demonicbliss-VirtualBox.51055 > _gateway.domain: 29562+ [1au] PTR? 12.1.168.192.in-addr.arpa. (54)
14:01:42.404966 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.51055: 29562 NXDomain* 0/1/1 (109)
14:01:42.405286 enp0s3 Out IP demonicbliss-VirtualBox.51055 > _gateway.domain: 29562+ PTR? 12.1.168.192.in-addr.arpa. (43)
14:01:42.410827 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.51055: 29562 NXDomain* 0/1/0 (98)
14:01:42.411153 lo    In  IP localhost.domain > localhost.35638: 40232*$ 2/0/1 PTR demonicbliss-VirtualBox., PTR demonicbliss-Virtua
lBox.local. (134)
14:01:47.297047 enp0s3 B   ARP, Request who-has demonicbliss-VirtualBox tell _gateway, length 46
14:01:47.297085 enp0s3 Out ARP, Reply demonicbliss-VirtualBox is-at 08:00:27:67:59:45 (oui Unknown), length 28
14:01:47.421301 enp0s3 Out ARP, Request who-has _gateway tell demonicbliss-VirtualBox, length 28
14:01:47.436343 enp0s3 In  ARP, Reply _gateway is-at b4:3d:08:59:9f:98 (oui Unknown), length 46
14:01:52.209650 enp0s3 B   ARP, Request who-has _gateway tell 192.168.1.3, length 46
14:01:54.464257 enp0s3 M   IP 192.168.1.7.mdns > mdns.mcast.net.mdns: 106 [2q] PTR (QM)? _233637DE._sub._googlecast._tcp.local. PTR
(QM)? _googlecast._tcp.local. (61)
14:01:54.472686 lo    In  IP localhost.55290 > localhost.domain: 27863+ [1au] PTR? 251.0.0.224.in-addr.arpa. (53)
14:01:54.473393 enp0s3 Out IP demonicbliss-VirtualBox.36802 > _gateway.domain: 48138+ [1au] PTR? 251.0.0.224.in-addr.arpa. (53)
14:01:54.480115 enp0s3 In  IP _gateway.domain > demonicbliss-VirtualBox.36802: 48138 1/4/9 PTR mdns.mcast.net. (344)
14:01:54.481055 lo    In  IP localhost.domain > localhost.55290: 27863 1/4/9 PTR mdns.mcast.net. (344)
```

Observation

Step 3: Understand the output format.

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

```
root@demonicbliss-VirtualBox:~# sudo tcpdump -i any -c5 icmp
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
14:05:13.582097 enp0s3 Out IP demonicbliss-VirtualBox > maa03s41-in-f14.1e100.net: ICMP echo request, id 3, seq 1, length 64
14:05:13.594333 enp0s3 In  IP maa03s41-in-f14.1e100.net > demonicbliss-VirtualBox: ICMP echo reply, id 3, seq 1, length 64
14:05:14.583643 enp0s3 Out IP demonicbliss-VirtualBox > maa03s41-in-f14.1e100.net: ICMP echo request, id 3, seq 2, length 64
14:05:14.594154 enp0s3 In  IP maa03s41-in-f14.1e100.net > demonicbliss-VirtualBox: ICMP echo reply, id 3, seq 2, length 64
14:05:15.584391 enp0s3 Out IP demonicbliss-VirtualBox > maa03s41-in-f14.1e100.net: ICMP echo request, id 3, seq 3, length 64
5 packets captured
6 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

```
root@demonicbliss-VirtualBox:~# sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
14:08:40.954276 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.55050 > 2001:67c:1562::24.80: Flags [S], seq 404728789, win 64800, opti
mss 1440,sackOK,TS val 1781747065 ecr 0,nop,wscale 7], length 0
`....(.@............u-j. ..|.b.........$.
.P........... .R.........
j3My........
14:08:40.956997 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.60992 > 2620:2d:4000:1::2b.80: Flags [S], seq 389368723, win 64800, opt
[mss 1440,sackOK,TS val 1854537876 ecr 0,nop,wscale 7], length 0
`./..(.@............u-j.& .-@..........+.@.P.5K........ ...........
n...........
14:08:40.959138 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.38362 > 2620:2d:4000:1::22.80: Flags [S], seq 2740494923, win 64800, op
 [mss 1440,sackOK,TS val 2476177841 ecr 0,nop,wscale 7], length 0
`
-}.(.@............u-j.& .-@..........."...P.X.K....... ...........
..y.........
14:08:40.965571 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.59986 > 2620:2d:4000:1::23.80: Flags [S], seq 359319670, win 64800, opt
[mss 1440,sackOK,TS val 2319106911 ecr 0,nop,wscale 7], length 0
`....(.@............u-j.& .-@..........#.R.P.j.v....... ...........
.:._........
14:08:40.969127 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.38928 > 2620:2d:4000:1::2a.80: Flags [S], seq 3802992839, win 64800, op
 [mss 1440,sackOK,TS val 2725373745 ecr 0,nop,wscale 7], length 0
`.s..(.@............u-j.& .-@..........*...P........... ...........
.q.1........
14:08:40.976085 enp0s3 Out IP6 fe80::419:c9d9:752d:6a80.40910 > 2001:67c:1562::23.80: Flags [S], seq 1805952234, win 64800, opt
[mss 1440,sackOK,TS val 568056911 ecr 0,nop,wscale 7], length 0
`.B..(.@............u-j. ..|.b.........#...Pk........... .Q.........
!..O........
14:08:50.175434 enp0s3 Out IP 192.168.1.12.46248 > 142.250.195.163.80: Flags [S], seq 1774662459, win 64240, options [mss 1460,
K,TS val 3394249439 ecr 0,nop,wscale 7], length 0
E..<.n@.@.............Pi.3;...................
.P".........
```

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

```
demonicbliss@demonicbliss-VirtualBox:~$ 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
10 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

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  9.557 ms  9.425 ms  9.359 ms
 2  * * *
 3  202.88.156.197 (202.88.156.197)  9.128 ms  9.069 ms  9.011 ms
 4  * * *
 5  10.240.254.100 (10.240.254.100)  8.783 ms  8.726 ms  8.757 ms
 6  * 10.240.254.1 (10.240.254.1)  3.889 ms  3.724 ms
 7  10.241.1.1 (10.241.1.1)  10.101 ms  10.056 ms  10.010 ms
 8  * * *
 9  142.250.172.12 (142.250.172.12)  22.051 ms  18.988 ms  18.551 ms
10  * * *
11  108.170.231.130 (108.170.231.130)  27.942 ms 74.125.242.129 (74.125.242.129)  27.791 ms 142.250.235.104 (142.250.235.104)  27.73
0 ms
12  74.125.242.154 (74.125.242.154)  11.493 ms 74.125.242.138 (74.125.242.138)  21.216 ms 74.125.242.147 (74.125.242.147)  20.707 ms
13  * 108.170.253.97 (108.170.253.97)  19.798 ms *
14  142.251.55.29 (142.251.55.29)  19.568 ms * 142.251.55.31 (142.251.55.31)  19.459 ms
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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

destination address: 142.250.196.36

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

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets
 1  192.168.1.1  4.549 ms  11.113 ms  11.053 ms
 2  * * *
 3  202.88.156.197  10.840 ms  10.786 ms  10.712 ms
 4  * * *
 5  10.240.254.100  10.498 ms  10.441 ms  10.382 ms
 6  * * *
 7  10.241.1.1  9.536 ms  9.151 ms  8.751 ms
 8  * * *
 9  142.250.172.12  12.296 ms  11.221 ms  11.111 ms
10  * * *
11  142.250.228.186  18.494 ms 142.250.233.144  15.410 ms 142.251.55.240  11.128 ms
12  74.125.242.131  11.756 ms 142.251.55.31  14.923 ms 74.125.242.139  12.509 ms
13  108.170.253.113  11.950 ms * *
14  142.251.55.31  13.799 ms  14.432 ms 142.251.55.29  14.389 ms
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  *^[[1;5D * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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

sudo traceroute -I www.google.com

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  4.422 ms  4.311 ms  4.213 ms
 2  * * *
 3  202.88.156.197 (202.88.156.197)  4.352 ms  4.317 ms  4.280 ms
 4  192.168.102.10 (192.168.102.10)  19.824 ms * *
 5  142.250.172.12 (142.250.172.12)  19.727 ms  23.083 ms  23.039 ms
 6  216.239.43.131 (216.239.43.131)  19.572 ms  21.016 ms  30.859 ms
 7  142.251.55.29 (142.251.55.29)  19.090 ms  18.801 ms  18.696 ms
 8  maa03s45-in-f4.1e100.net (142.250.196.36)  18.170 ms  17.655 ms  17.395 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

```
demonicbliss@demonicbliss-VirtualBox:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  10.162 ms  9.849 ms  9.755 ms
 2  * * *
 3  202.88.156.197 (202.88.156.197)  9.514 ms  9.466 ms  9.415 ms
 4  10.241.1.6 (10.241.1.6)  9.356 ms * *
 5  10.240.254.50 (10.240.254.50)  9.338 ms  9.141 ms  9.222 ms
 6  * 10.240.254.1 (10.240.254.1)  9.857 ms *
 7  10.241.1.1 (10.241.1.1)  8.952 ms  8.395 ms  8.115 ms
 8  * * *
 9  142.250.172.12 (142.250.172.12)  15.550 ms  14.360 ms  13.297 ms
10  216.239.43.131 (216.239.43.131)  13.395 ms  10.601 ms  13.061 ms
11  142.251.55.29 (142.251.55.29)  12.977 ms  12.448 ms 142.251.55.31 (142.251.55.31)  15.588 ms
12  maa03s45-in-f4.1e100.net (142.250.196.36)  13.232 ms  9.886 ms  13.536 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

```
demonicbliss@demonicbliss-VirtualBox:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-22 20:15 CET
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.036s 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 4.46 seconds
```

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

nmap 163.53.78.128

```
demonicbliss@demonicbliss-VirtualBox:~$ nmap 163.58.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-22 20:16 CET
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 3.03 seconds
```

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

nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
demonicbliss@demonicbliss-VirtualBox:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-22 20:18 CET
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0032s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds

Nmap done: 3 IP addresses (1 host up) scanned in 9.86 seconds
```

Submission:

Students are expected to take the screenshot of results - after execution of every command in every task.
They are expected to write the Task and 2-3 lines of their observation followed by screenshots.
Submissions will be through google forms.