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 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, 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**

```
root@UbuntuOS:/home/ashhad# 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::69e4:ea35:6f22:6b04  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:67:1c:97  txqueuelen 1000  (Ethernet)
        RX packets 288524  bytes 407198074 (407.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 44466  bytes 5041696 (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 9758  bytes 1112229 (1.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9758  bytes 1112229 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Analyze and fill the following table:

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address	
enp0s3	10.0.2.15/24	08:00:27:67:1c:97	
lo	127.0.0.1/8	00:00:00:00:00:00	

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**

```
root@UbuntuOS:/home/ashhad# ifconfig enp0s3 10.0.2.107 255.255.255.0
root@UbuntuOS:/home/ashhad# 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 08:00:27:67:1c:97 brd ff:ff:ff:ff:ff:ff
    inet 255.255.255.0/32 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::69e4:ea35:6f22:6b04/64 scope link noprefixroute
```

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

sudo ifconfig interface_name down sudo

ifconfig interface_name up

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

ip neigh

```
root@UbuntuOS:/home/ashhad# ifconfig enp0s3 down
root@UbuntuOS:/home/ashhad# ifconfig enp0s3 up
root@UbuntuOS:/home/ashhad# ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 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: Analyze the following in Terminal

- TTL
- Protocol used by ping
- Time

Step 5: Analyze the following in Wireshark

```
root@UbuntuOS:/home/ashhad# ping 10.0.2.107
PING 10.0.2.107 (10.0.2.107) 56(84) bytes of data.
From 10.0.2.15 icmp_seq=1 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=2 Destination Host Unreachable
From 10.0.2.15 icmp_seq=3 Destination Host Unreachable
From 10.0.2.15 icmp_seq=5 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=6 Destination Host Unreachable
From 10.0.2.15 icmp_seq=7 Destination Host Unreachable
From 10.0.2.15 icmp_seq=9 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=10 Destination Host Unreachable
From 10.0.2.15 icmp_seq=11 Destination Host Unreachable
From 10.0.2.15 icmp_seq=13 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=14 Destination Host Unreachable
From 10.0.2.15 icmp_seq=15 Destination Host Unreachable
From 10.0.2.15 icmp_seq=17 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=18 Destination Host Unreachable
From 10.0.2.15 icmp_seq=19 Destination Host Unreachable
From 10.0.2.15 icmp_seq=21 Destination Host Unreachable
ping: sendmsg: No route to host
From 10.0.2.15 icmp_seq=22 Destination Host Unreachable
From 10.0.2.15 icmp_seq=23 Destination Host Unreachable
From 10.0.2.15 icmp_seq=25 Destination Host Unreachable
From 10.0.2.15 icmp_seq=26 Destination Host Unreachable
From 10.0.2.15 icmp_seq=27 Destination Host Unreachable
^Z
[3]+  Stopped                 ping 10.0.2.107
```

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	480	487
Source IP address	162.247.241.14	10.0.2.15
Destination IP address	10.0.2.15	162.247.241.14
ICMP Type Value	2	2
ICMP Code Value	2	2
Source Ethernet Address	52:54:00:12:35:02	08:00:27:67:1c:97
Destination Ethernet Address	08:00:27:67:1c:97	52:54:00:12:35:02
Internet Protocol Version	4	4

Time To Live (TTL) Value	72	72

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

Details	First Echo Request	First Echo Reply
Frame Number	4943	4947
Source Port	45724	80
Destination Port	80	45724
Source IP address	10.0.2.15	91.189.91.49
Destination IP address	91.189.91.49	10.0.2.15
Source Ethernet Address	08:00:27:67:1c:97	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	08:00:27:67:1c:97

tc

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

HTTP Request		HTTP Response	
Get	HTTP/1.1\r\n	Server	openrest\r\n
host	connectivity-check.ubuntu.com\r\n	Content-Type	Text/html
User-Agent	Microsoft-CryptoAPI/10.0	Date	16 Jan 2024
Accept-Language	en-U	Location	India
Accept-Encoding	gzip, deflate\r\n	Content-Length	22/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

```
root@UbuntuOS:/home/ashhad# tcpdump -D
1.enp0s3 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
```

sudo

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

sudo tcpdump -i any

```
root@UbuntuOS:/home/ashhad# 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
22:44:21.813605 lo    In  IP localhost.33793 > localhost.domain: 49756+ [1au] A? bam.nr-data.net. (44)
22:44:21.813621 lo    In  IP localhost.33793 > localhost.domain: 21592+ [1au] AAAA? bam.nr-data.net. (44)
22:44:21.814772 lo    In  IP localhost.domain > localhost.33793: 49756 3/0/1 CNAME bam.cell.nr-data.net., CNAME fastly-tls12-bam.nr-data.net., A 162.247.243.29 (114)
22:44:21.815087 lo    In  IP localhost.domain > localhost.33793: 21592 3/0/2 CNAME bam.cell.nr-data.net., CNAME fastly-tls12-bam.nr-data.net., AAAA ::ffff:162.247.243.29 (142)
22:44:21.819858 lo    In  IP localhost.44477 > localhost.domain: 65299+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
22:44:21.821524 lo    In  IP localhost.domain > localhost.44477: 65299*$ 1/0/1 PTR localhost. (75)
22:44:21.826020 enp0s3 Out IP UbuntuOS.38234 > 162.247.243.29.https: Flags [P.], seq 2750487137:2750488142, ack 747909021, win 63784, length 1005
22:44:21.826554 enp0s3 Out IP UbuntuOS.57586 > 162.247.243.29.https: Flags [P.], seq 1390575094:1390576071, ack 761152848, win 63784, length 977
22:44:21.826556 lo    In  IP localhost.50207 > localhost.domain: 60454+ [1au] A? bam.nr-data.net. (44)
22:44:21.827589 lo    In  IP localhost.50207 > localhost.domain: 43811+ [1au] AAAA? bam.nr-data.net. (44)
22:44:21.829530 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.38234: Flags [.], ack 1005, win 65535, length 0
22:44:21.829530 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.57586: Flags [.], ack 977, win 65535, length 0
22:44:21.830016 lo    In  IP localhost.domain > localhost.50207: 60454 3/0/1 CNAME bam.cell.nr-data.net., CNAME fastly-tls12-bam.nr-data.net., A 162.247.243.29 (114)
22:44:21.830268 lo    In  IP localhost.domain > localhost.50207: 43811 3/0/2 CNAME bam.cell.nr-data.net., CNAME fastly-tls12-bam.nr-data.net., AAAA ::ffff:162.247.243.29 (142)
22:44:21.944732 lo    In  IP localhost.48081 > localhost.domain: 23269+ [1au] PTR? 29.243.247.162.in-addr.arpa. (56)
22:44:21.946242 enp0s3 Out IP UbuntuOS.32994 > 192.168.82.154.domain: 25138+ PTR? 29.243.247.162.in-addr.arpa. (45)
22:44:22.004940 enp0s3 In  IP 192.168.82.154.domain > UbuntuOS.32994: 25138 NXDomain 0/1/0 (110)
22:44:22.006370 lo    In  IP localhost.domain > localhost.48081: 23269 NXDomain 0/1/1 (121)
22:44:22.009683 lo    In  IP localhost.34579 > localhost.domain: 58664+ [1au] PTR? 15.2.0.10.in-addr.arpa. (51)
22:44:22.010039 enp0s3 Out IP UbuntuOS.45112 > 192.168.82.154.domain: 2616+ PTR? 15.2.0.10.in-addr.arpa. (40)
22:44:22.024603 enp0s3 In  IP 192.168.82.154.domain > UbuntuOS.45112: 2616 NXDomain 0/0/0 (40)
22:44:22.025969 lo    In  IP localhost.domain > localhost.34579: 58664*$ 2/0/1 PTR UbuntuOS., PTR UbuntuOS.local. (101)
22:44:22.054250 lo    In  IP localhost.38489 > localhost.domain: 30366+ [1au] PTR? 154.82.168.192.in-addr.arpa. (56)
22:44:22.059484 enp0s3 Out IP UbuntuOS.46527 > 192.168.82.154.domain: 63791+ PTR? 154.82.168.192.in-addr.arpa. (45)
22:44:22.119443 enp0s3 In  IP 192.168.82.154.domain > UbuntuOS.46527: 63791 NXDomain* 0/1/0 (104)
22:44:22.119444 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.38234: Flags [P.], seq 1:400, ack 1005, win 65535, length 399
22:44:22.120887 lo    In  IP localhost.domain > localhost.38489: 30366 NXDomain 0/1/1 (115)
22:44:22.125718 enp0s3 Out IP UbuntuOS.38234 > 162.247.243.29.https: Flags [.], ack 400, win 63784, length 0
22:44:22.119444 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.38234: Flags [F.], seq 400, ack 1005, win 65535, length 0
22:44:22.119444 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.57586: Flags [P.], seq 1:388, ack 977, win 65535, length 387
22:44:22.126433 enp0s3 Out IP UbuntuOS.57586 > 162.247.243.29.https: Flags [.], ack 388, win 63784, length 0
22:44:22.132224 enp0s3 Out IP UbuntuOS.38234 > 162.247.243.29.https: Flags [P.], seq 1005:1029, ack 401, win 63784, length 24
22:44:22.133153 enp0s3 Out IP UbuntuOS.38234 > 162.247.243.29.https: Flags [F.], seq 1029, ack 401, win 63784, length 0
22:44:22.143588 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.38234: Flags [.], ack 1029, win 65535, length 0
22:44:22.143589 enp0s3 In  IP 162.247.243.29.https > UbuntuOS.38234: Flags [.], ack 1030, win 65535, length 0
```

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

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

```
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
22:10:55.622976 enp0s3 In  IP _gateway > UBUNTU: ICMP net 96.10.190.35.bc.googleusercontent.com unreachable, length 36
22:10:55.622976 enp0s3 In  IP _gateway > UBUNTU: ICMP net 96.10.190.35.bc.googleusercontent.com unreachable, length 36
22:12:56.709857 enp0s3 In  IP _gateway > UBUNTU: ICMP net maa03s29-in-f2.1e100.net unreachable, length 36
22:12:56.709858 enp0s3 In  IP _gateway > UBUNTU: ICMP net maa03s29-in-f2.1e100.net unreachable, length 36
22:12:56.981900 enp0s3 In  IP _gateway > UBUNTU: ICMP net a23-200-49-86.deploy.static.akamaitechnologies.com unreachable, length 3
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@UbuntuOS:/home/ashhad# 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
23:35:34.067322 enp0s3 Out IP 10.0.2.15.34366 > 104.18.20.226.80: Flags [.], ack 200645813, win 62780, length 0
E..(l.@.@.E/
...h....>.P..M.....P..<....
23:35:34.096182 enp0s3 In  IP 104.18.20.226.80 > 10.0.2.15.34366: Flags [.], ack 1, win 65535, length 0
E..(a
..@...h...
....P.>......M.P.............
23:35:37.645481 enp0s3 Out IP 10.0.2.15.34376 > 104.18.20.226.80: Flags [.], ack 200899876, win 62780, length 0
E..(..@.@...
...h....H.P.S]Y..}$P..<....
23:35:37.645635 enp0s3 Out IP 10.0.2.15.34364 > 104.18.20.226.80: Flags [.], ack 200707876, win 62780, length 0
E..(B.@.@.oL
...h....<.P..k....$P..<....
23:35:37.647667 enp0s3 In  IP 104.18.20.226.80 > 10.0.2.15.34376: Flags [.], ack 1, win 65535, length 0
E..(a...@...h...
....P.H..}$.S]ZP....n........
23:35:37.647687 enp0s3 In  IP 104.18.20.226.80 > 10.0.2.15.34364: Flags [.], ack 1, win 65535, length 0
E..(a...@...h...
....P.<...$..k.P.............
23:35:39.934408 enp0s3 Out IP 10.0.2.15.34078 > 104.18.20.226.80: Flags [.], ack 205636239, win 63540, length 0
E..(..@.@...
...h......P...:.A..P..4....
23:35:39.935227 enp0s3 In  IP 104.18.20.226.80 > 10.0.2.15.34078: Flags [.], ack 1, win 65535, length 0
E..(a...@...h...
....P...A.....;P....Z........
23:35:40.410990 enp0s3 Out IP 10.0.2.15.43788 > 18.161.210.238.80: Flags [.], ack 206656945, win 63297, length 0
E..(..@.@.C.
..........P.....QU.P..A....
23:35:40.411174 enp0s3 Out IP 10.0.2.15.43772 > 18.161.210.238.80: Flags [.], ack 206592946, win 63296, length 0
E..(..@.@.W.
..........PX.z..P[.P..@....
10 packets captured
12 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:

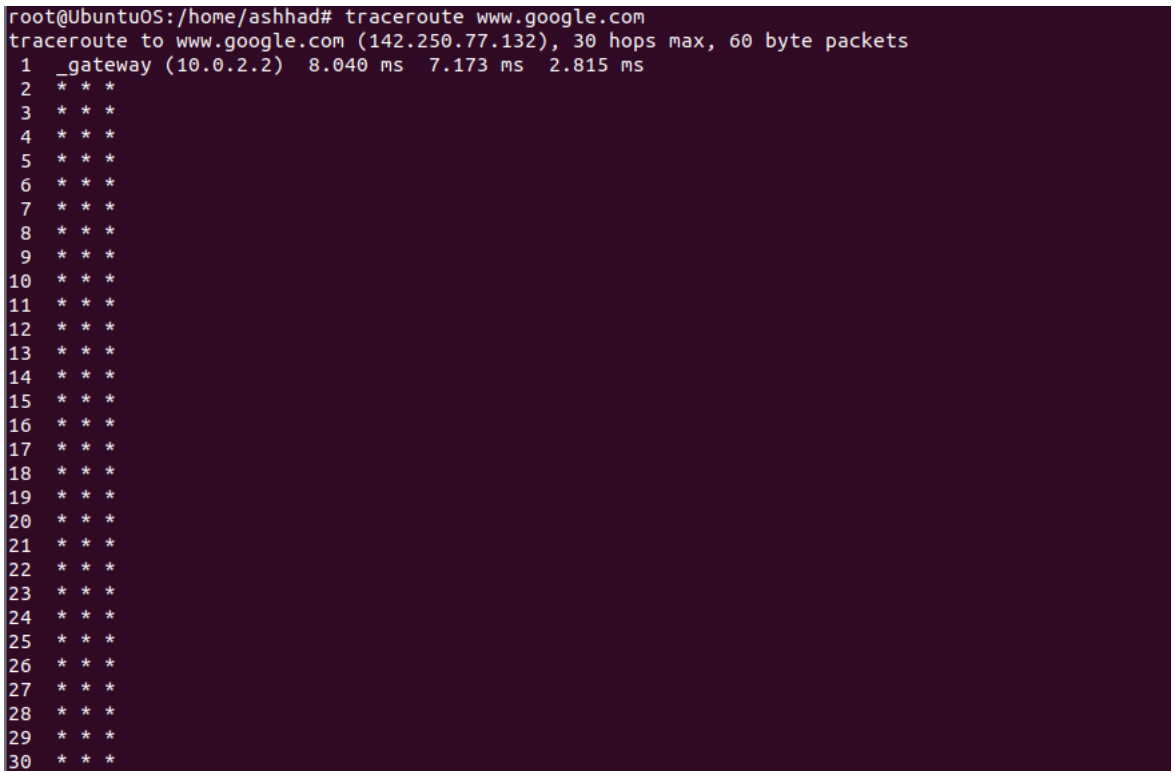
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80

```
root@UbuntuOS:/home/ashhad# 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
12 packets received by filter
0 packets dropped by kernel
```

Task 5: Perform Traceroute checks

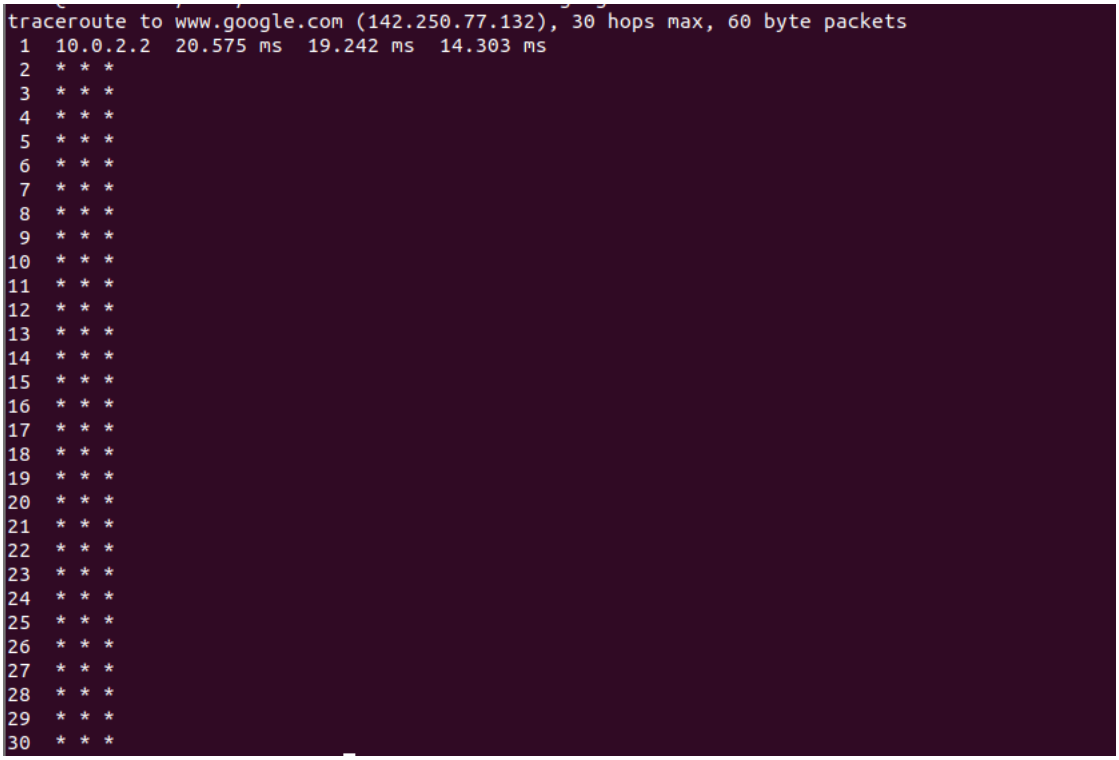
Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com



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**



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

sudo traceroute -I www.google.com trace

```
root@UbuntuOS:/home/ashhad# traceroute -I www.google.com
traceroute to www.google.com (142.250.77.132), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  13.186 ms  12.459 ms  11.614 ms
 2  192.168.82.154 (192.168.82.154)  79.856 ms  78.911 ms  75.595 ms
 3  * * *
 4  * * *
 5  * * *
 6  192.168.225.146 (192.168.225.146)  91.704 ms  58.917 ms  63.584 ms
 7  192.168.151.229 (192.168.151.229)  57.634 ms  25.863 ms 192.168.151.225 (192.168.151.225)  24.944 ms
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  72.14.217.252 (72.14.217.252)  62.027 ms  74.502 ms  74.405 ms
14  142.251.227.211 (142.251.227.211)  74.300 ms  74.258 ms  74.202 ms
15  142.251.55.207 (142.251.55.207)  60.169 ms  60.103 ms  60.064 ms
16  maa05s16-in-f4.1e100.net (142.250.77.132)  35.140 ms  33.074 ms  36.788 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

```
root@UbuntuOS:/home/ashhad# traceroute -T www.google.com
traceroute to www.google.com (142.250.77.132), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.560 ms  0.365 ms  0.257 ms
 2  maa05s16-in-f4.1e100.net (142.250.77.132)  38.720 ms  38.629 ms  44.705 ms
root@UbuntuOS:/home/ashhad#
```

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

```
root@UbuntuOS:/home/ashhad#  nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-16 22:20 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.013s latency).
Other addresses for www.pes.edu (not scanned): 64:ff9b::34ac:ccc4
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 5.93 seconds
```

Step 2: Alternatively, use an IP address to scan

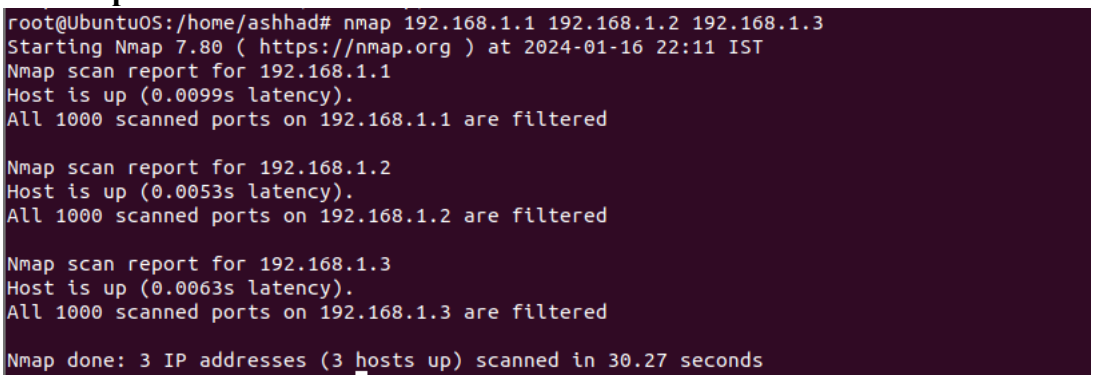
. **nmap 163.53.78.128**

```
root@UbuntuOS:/home/ashhad# nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-16 22:02 IST
Nmap scan report for 163.53.78.128
Host is up (0.012s latency).
All 1000 scanned ports on 163.53.78.128 are filtered

Nmap done: 1 IP address (1 host up) scanned in 6.04 seconds
```

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

nmap 192.168.1.1 192.168.1.2 192.168.1.3



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.

Questions on above observations: (Optional)

1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?
2) When was the HTML file that you are retrieving last modified at the server?
3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?
4) How will you identify remote host apps and OS?