

## **Week #1**

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

NAME = RACHAPPA

SRN = PES1UG19CS359

ROLL NO = 1

### **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:

```

rachappa@balaji:~$ 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::835d:883e:3f1c:a0b5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:37:41:ee txqueuelen 1000 (Ethernet)
    RX packets 199 bytes 240941 (240.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 151 bytes 16957 (16.9 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 128 bytes 10895 (10.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 128 bytes 10895 (10.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

**ip address table:**

Interface name	IP address (IPv4 / IPv6)	MAC address	
Enp0s3	10.0.2.15	08-00-27-37-41-ee	
lo	127.0.0.1		

**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@balaji:/home/rachappa# sudo ifconfig enp0s3 10.0.1.1
root@balaji:/home/rachappa# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.1 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::835d:883e:3f1c:a0b5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:37:41:ee txqueuelen 1000 (Ethernet)
    RX packets 260 bytes 283022 (283.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 214 bytes 31295 (31.2 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 137 bytes 11660 (11.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 137 bytes 11660 (11.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

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

**sudo ifconfig interface\_name down**

```

root@balaji:/home/rachappa# sudo ifconfig enp0s3 down
root@balaji:/home/rachappa# 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 138 bytes 11733 (11.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11733 (11.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@balaji:/home/rachappa# █

```

**sudo ifconfig interface\_name up**

```
root@balaji:/home/rachappa# sudo ifconfig enp0s3 up
root@balaji:/home/rachappa# 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::835d:883e:3f1c:a0b5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:37:41:ee txqueuelen 1000 (Ethernet)
    RX packets 268 bytes 284424 (284.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 243 bytes 34775 (34.7 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 138 bytes 11733 (11.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11733 (11.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

**ip neigh**

```
root@balaji:/home/rachappa# ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
root@balaji:/home/rachappa#
```

## **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



```
root@balaji:/home/rachappa# wireshark
** (wireshark:2135) 22:17:09.106465 [GUI WARNING] -- QStandardPaths: runtime directory
'/run/user/1001' is not owned by UID 0, but a directory permissions 0700 owned by UID
1001 GID 1001
** (wireshark:2135) 22:17:09.244805 [GUI WARNING] -- QStandardPaths: runtime directory
'/run/user/1001' is not owned by UID 0, but a directory permissions 0700 owned by UID
1001 GID 1001
** (wireshark:2135) 22:17:17.376361 [Capture MESSAGE] -- Capture Start ...
** (wireshark:2135) 22:17:17.435255 [Capture MESSAGE] -- Capture started
** (wireshark:2135) 22:17:17.435334 [Capture MESSAGE] -- File: "/tmp/wireshark_anyFUCD
Z1.pcapng"
```

**Step 3:** In terminal, type **ping 10.0.your\_section.your\_sno**

```
rachappa@balaji:~$ ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=0.022 ms
64 bytes from 10.0.1.1: icmp_seq=5 ttl=64 time=0.031 ms
64 bytes from 10.0.1.1: icmp_seq=6 ttl=64 time=0.025 ms
64 bytes from 10.0.1.1: icmp_seq=7 ttl=64 time=0.027 ms
64 bytes from 10.0.1.1: icmp_seq=8 ttl=64 time=0.028 ms
64 bytes from 10.0.1.1: icmp_seq=9 ttl=64 time=0.027 ms
64 bytes from 10.0.1.1: icmp_seq=10 ttl=64 time=0.028 ms
64 bytes from 10.0.1.1: icmp_seq=11 ttl=64 time=0.028 ms
64 bytes from 10.0.1.1: icmp_seq=12 ttl=64 time=0.028 ms
64 bytes from 10.0.1.1: icmp_seq=13 ttl=64 time=0.028 ms
64 bytes from 10.0.1.1: icmp_seq=14 ttl=64 time=0.026 ms
64 bytes from 10.0.1.1: icmp_seq=15 ttl=64 time=0.027 ms
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=1/256, ttl=64
2	0.000022291	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=1/256, ttl=64
3	1.007601551	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=2/512, ttl=64
4	1.007608572	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=2/512, ttl=64
5	2.030951359	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=3/768, ttl=64
6	2.030989986	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=3/768, ttl=64
7	3.055584194	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=4/1024, ttl=64
8	3.055589884	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64
9	4.080453615	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=5/1280, ttl=64
10	4.080460451	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=5/1280, ttl=64
11	5.102804866	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=6/1536, ttl=64
12	5.102811488	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=6/1536, ttl=64
13	6.127405931	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=7/1792, ttl=64
14	6.127412328	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=7/1792, ttl=64
15	7.151195612	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=8/2048, ttl=64
16	7.151202582	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=8/2048, ttl=64
17	8.175398693	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) request id=0x0001, seq=9/2304, ttl=64
18	8.175405585	10.0.1.1	10.0.1.1	ICMP	100	Echo (ping) reply id=0x0001, seq=9/2304, ttl=64

▶ Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0  
 ▶ Linux cooked capture v1  
 ▶ Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.1  
 ▶ Internet Control Message Protocol

## Observations to be made

### Step 4: Analyze the following in Terminal

- TTL

TTL IS TTL IS 64

- Protocol used by ping

PING PROTOCOL WORK ON ICMP

- Time

Rtt min/avg/max/mdev = 0.022/0.035/0.119/0.019

### Step 5: Analyze the following in Wireshark

Frame 1



Wireshark · Packet 2 · any

File Edit

Apply

No. T

1 0.

2 0.

3 1.

4 1.

5 2.

6 2.

7 3.

8 3.

9 4.

10 4.

11 5.

12 5.

13 6.

14 6.

15 7.

16 7.

17 8.

18 8.

Frame 2: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0

Linux cooked capture v1

Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.1

Internet Control Message Protocol

0000 00 00 03 04 00 06 00 00 00 00 00 00 00 00 00 .....  
0010 45 00 00 54 1e c6 00 00 40 01 45 e2 0a 00 01 01 E..T... @.E....  
0020 0a 00 01 01 00 00 06 52 00 01 00 01 1e 00 d4 63 .....R.....c  
0030 00 00 00 00 3d 75 0b 00 00 00 00 00 10 11 12 13 ....=u.....  
0040 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 .....!"#  
0050 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 \$%&'()\*+,-./0123  
0060 34 35 36 37 4567

Help

Close

1/256, ttl=64  
1/256, ttl=64  
2/512, ttl=64  
2/512, ttl=64  
3/768, ttl=64  
3/768, ttl=64  
4/1024, ttl=64  
4/1024, ttl=64  
5/1280, ttl=64  
5/1280, ttl=64  
6/1536, ttl=64  
6/1536, ttl=64  
7/1792, ttl=64  
7/1792, ttl=64  
8/2048, ttl=64  
8/2048, ttl=64  
9/2304, ttl=64  
9/2304, ttl=64



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

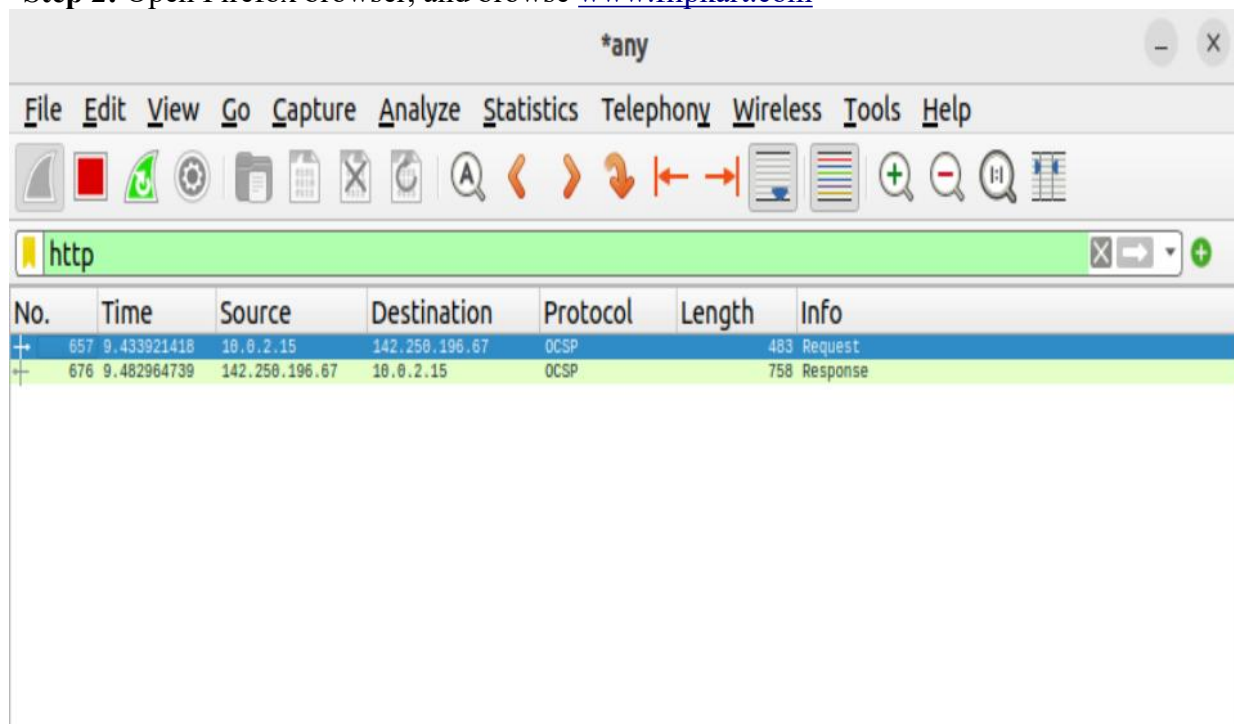
Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.1.1	10.0.1.1
Destination IP address	10.0.1.1	10.0.1.1
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	10.0.1.1	10.0.1.1
Destination Ethernet Address	10.0.1.1	10.0.1.1
Internet Protocol Version	4	4
Time 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](http://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	657	676
Source Port	60892	80
Destination Port	180	60892
Source IP address	10.0.2.15	142.250.196.67
Destination IP address	142.250.196.67	10.0.2.15
Source Ethernet Address	10.0.2.15	142.250.196.67
Destination Ethernet Address	142.250.196.67	10.0.2.15

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

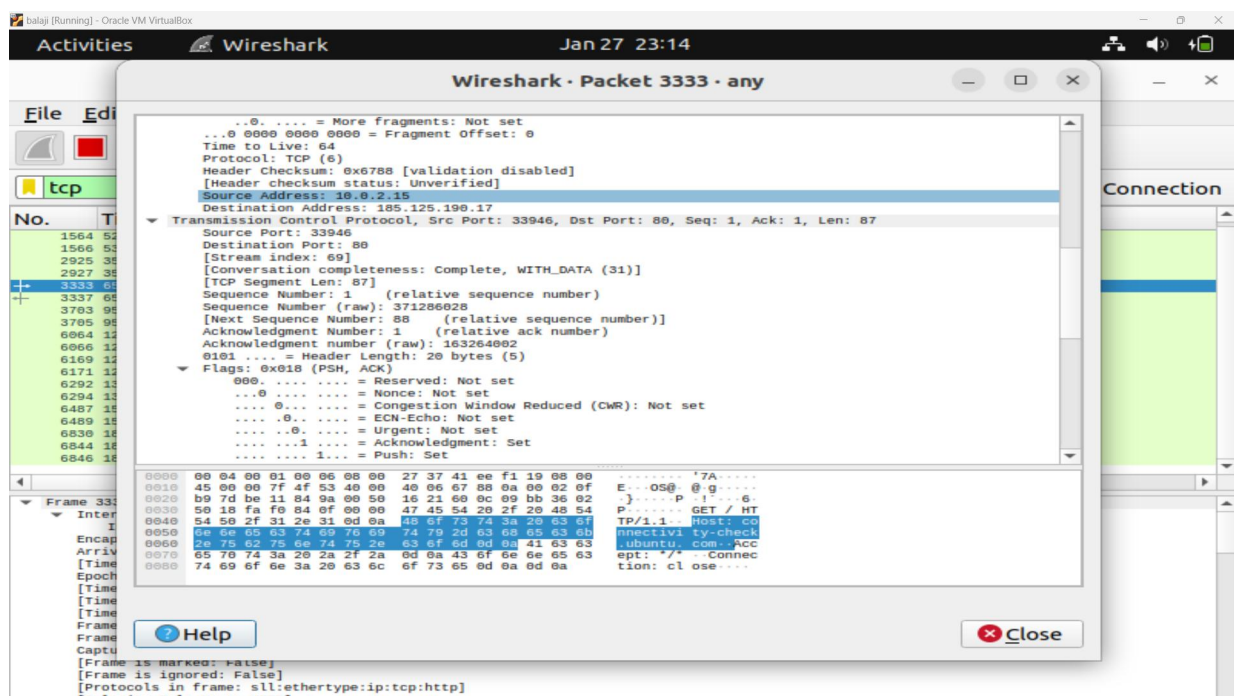
HTTP Request		HTTP Response	
Get		Server	

Host	Ocsp.pki.goog \r\n	Content-Type	Application/osc p-request
User-Agent	Mozilla/5.0	Date	Jan 27 2023- 22
Accept-Language	En-us	Location	
Accept-Encoding	Gzip,deflate	Content-Length	472\r\n
Connection	Keep-alive	Connection	

## Using Wireshark's Follow TCP Stream

**Step 1:** Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'. 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@balaji: /home/rachappa
root@balaji:/home/rachappa# sudo tcpdump -D
. enp0s3 [Up, Running, Connected]
. any (Pseudo-device that captures on all interfaces) [Up, Running]
. lo [Up, Running, Loopback]
. bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
. nflog (Linux netfilter log (NFLOG) interface) [none]
. nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
. dbus-system (D-Bus system bus) [none]
. dbus-session (D-Bus session bus) [none]
root@balaji:/home/rachappa#
```

**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](http://www.google.com) in browser.



```
balaji [Running] - Oracle VM VirtualBox
Activities Terminal Jan 27 23:18
root@balaji: /home/rachappa

root@balaji: /home/rachappa x rachappa@balaji: ~

root@balaji:/home/rachappa# 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
23:18:24.397798 lo In IP balaji > balaji: ICMP echo request, id 3, seq 4, length 64
23:18:24.397820 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 4, length 64
23:18:24.470437 lo In IP localhost.45364 > localhost.domain: 41726+ [Iau] PTR? 15.2.0.10.in-addr.arpa. (51)
23:18:24.478721 enp0s3 Out IP balaji.38256 > 183.82.243.66.actcorp.in.domain: 51235+ [Iau] PTR? 15.2.0.10.in-addr.arpa. (51)
23:18:24.484908 enp0s3 In IP 183.82.243.66.actcorp.in.domain > balaji.38256: 51235 NXDomain 0/1/1 (128)
23:18:24.485009 enp0s3 Out IP balaji.38256 > 183.82.243.66.actcorp.in.domain: 51235+ PTR? 15.2.0.10.in-addr.arpa. (40)
23:18:24.499000 enp0s3 In IP 183.82.243.66.actcorp.in.domain > balaji.38256: 51235 NXDomain 0/1/0 (117)
23:18:24.499222 lo In IP localhost.domain > localhost.45364: 41726+ 2/0/1 PTR balaji. PTR balaji.local. (97)
23:18:24.579028 lo In IP localhost.59959 > localhost.domain: 31299+ [Iau] PTR? 53.0.0.127.in-addr.arpa. (52)
23:18:24.579201 lo In IP localhost.domain > localhost.59959: 31299+ 1/0/1 PTR localhost. (75)
23:18:24.579515 lo In IP localhost.48282 > localhost.domain: 3286+ [Iau] PTR? 66.243.82.183.in-addr.arpa. (55)
23:18:24.579622 lo In IP localhost.domain > localhost.48282: 3286 1/0/1 PTR 183.82.243.66.actcorp.in. (93)
23:18:24.700011 enp0s3 Out IP balaji.36306 > 239.237.117.34.bc.googleusercontent.com.https: Flags [P.], seq 2017677408:2017677447, ack 329538278, win 64028, length 39
23:18:24.700315 enp0s3 In IP 239.237.117.34.bc.googleusercontent.com.https > balaji.36306: Flags [.], ack 39, win 65535, length 0
23:18:24.700338 enp0s3 Out IP balaji.36306 > 239.237.117.34.bc.googleusercontent.com.https: Flags [P.], seq 39:63, ack 1, win 64028, length 24
23:18:24.700402 enp0s3 Out IP balaji.36306 > 239.237.117.34.bc.googleusercontent.com.https: Flags [F.], seq 63, ack 1, win 64028, length 0
23:18:24.700719 enp0s3 In IP 239.237.117.34.bc.googleusercontent.com.https > balaji.36306: Flags [.], ack 63, win 65535, length 0
23:18:24.700719 enp0s3 In IP 239.237.117.34.bc.googleusercontent.com.https > balaji.36306: Flags [.], ack 64, win 65535, length 0
23:18:24.714778 enp0s3 In IP 239.237.117.34.bc.googleusercontent.com.https > balaji.36306: Flags [F.], seq 1, ack 64, win 65535, length 0
23:18:24.714797 enp0s3 Out IP balaji.36306 > 239.237.117.34.bc.googleusercontent.com.https: Flags [.], ack 2, win 64028, length 0
23:18:24.787722 lo In IP localhost.33577 > localhost.domain: 2560+ [Iau] PTR? 239.237.117.34.in-addr.arpa. (56)
23:18:24.788131 enp0s3 Out IP balaji.33196 > 183.82.243.66.actcorp.in.domain: 9879+ [Iau] PTR? 239.237.117.34.in-addr.arpa. (56)
23:18:24.817088 enp0s3 In IP 183.82.243.66.actcorp.in.domain > balaji.33196: 9879 1/0/1 PTR 239.237.117.34.bc.googleusercontent.com. (109)
23:18:24.817280 lo In IP localhost.domain > localhost.33577: 2560 1/0/1 PTR 239.237.117.34.bc.googleusercontent.com. (109)
23:18:25.425984 lo In IP balaji > balaji: ICMP echo request, id 3, seq 5, length 64
23:18:25.425990 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 5, length 64
23:18:25.446260 lo In IP balaji > balaji: ICMP echo request, id 3, seq 6, length 64
23:18:25.446284 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 6, length 64
23:18:27.471391 enp0s3 Out IP balaji.54610 > 162.247.241.14.https: Flags [.], ack 69280101, win 62780, length 0
23:18:27.471519 lo In IP balaji > balaji: ICMP echo request, id 3, seq 7, length 64
23:18:27.471557 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 7, length 64
23:18:27.471828 enp0s3 In IP 162.247.241.14.https > balaji.54610: Flags [.], ack 1, win 65535, length 0
23:18:27.506747 lo In IP localhost.34944 > localhost.domain: 18580+ [Iau] PTR? 14.241.247.162.in-addr.arpa. (56)
23:18:27.506971 enp0s3 Out IP balaji.51647 > 183.82.243.66.actcorp.in.domain: 36739+ [Iau] PTR? 14.241.247.162.in-addr.arpa. (56)
23:18:27.528901 enp0s3 In IP 183.82.243.66.actcorp.in.domain > balaji.51647: 36739 NXDomain 0/1/1 (121)
23:18:27.529049 enp0s3 Out IP balaji.51647 > 183.82.243.66.actcorp.in.domain: 36739+ PTR? 14.241.247.162.in-addr.arpa. (45)
23:18:27.551351 enp0s3 In IP 183.82.243.66.actcorp.in.domain > balaji.51647: 36739 NXDomain 0/1/0 (110)
23:18:27.551698 lo In IP localhost.domain > localhost.34944: 18580 NXDomain 0/1/1 (121)
^C
```

## 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**

```
balaji [Running] - Oracle VM VirtualBox
Activities Terminal Jan 27 23:19
root@balaji: /home/rachappa

root@balaji: /home/rachappa x rachappa@balaji: ~

root@balaji:/home/rachappa# 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
23:19:38.129744 lo In IP balaji > balaji: ICMP echo request, id 3, seq 76, length 64
23:19:38.129751 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 76, length 64
23:19:39.149820 lo In IP balaji > balaji: ICMP echo request, id 3, seq 77, length 64
23:19:39.149827 lo In IP balaji > balaji: ICMP echo reply, id 3, seq 77, length 64
23:19:40.175142 lo In IP balaji > balaji: ICMP echo request, id 3, seq 78, length 64
5 packets captured
12 packets received by filter
0 packets dropped by kernel
root@balaji:/home/rachappa#
```

**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@balaji:/home/rachappa# 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
23:21:46.809110 enp0s3 Out IP 10.0.2.15.47508 > 142.250.182.67.80: Flags [S], seq 18074
29992, win 64240, options [mss 1460,sackOK,TS val 2966105543 ecr 0,nop,wscale 7], lengt
h 0
E..<0.@@...
.....C...Pk.1h.....Q{.....
.....
23:21:46.820942 enp0s3 In IP 142.250.182.67.80 > 10.0.2.15.47508: Flags [S.], seq 3776
64001, ack 1807429993, win 65535, options [mss 1460], length 0
E..iX..@..'....C
....P.....k.1i'...'=.
23:21:46.821015 enp0s3 Out IP 10.0.2.15.47508 > 142.250.182.67.80: Flags [.], ack 1, wi
n 64240, length 0
E..(0.@@...
.....C...Pk.1i....P...Qg..
23:21:46.821128 enp0s3 Out IP 10.0.2.15.47508 > 142.250.182.67.80: Flags [P.], seq 1:42
8, ack 1, win 64240, length 427: HTTP: POST /gts1c3 HTTP/1.1
E...0.@@...
.....C...Pk.1i....P...S...POST /gts1c3 HTTP/1.1
Host: ojsp.pki.goog
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/10
9.0

```

```

Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 84
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

0R0P0N0L0J0      ..+.....y...a4...GB....$.c...t.....=...F..q5.'....,\...V1
.2.I..4
23:21:46.824971 enp0s3 In IP 142.250.182.67.80 > 10.0.2.15.47508: Flags [.], ack 428,
win 65535, length 0
E..(iY..@..*...C
....P.....k.3.P...=0.....
23:21:46.863162 enp0s3 In IP 142.250.182.67.80 > 10.0.2.15.47508: Flags [P.], seq 1:70
3, ack 428, win 65535, length 702: HTTP: HTTP/1.1 200 OK
E...ih..@..]...C
....P.....k.3.P....Z..HTTP/1.1 200 OK
Content-Type: application/ocsp-response
Date: Fri, 27 Jan 2023 17:51:46 GMT
Cache-Control: public, max-age=14400
Server: ocsp_responder
Content-Length: 472
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN

```

**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@balaji: /home/rachappa
root@balaji: /home/rachappa x rachappa@balaji: ~ x v
root@balaji:/home/rachappa# 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 2621
44 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel
root@balaji:/home/rachappa#

```

## Task 5: Perform Traceroute checks

**Step 1:** Run the traceroute using the following command.

**sudo traceroute [www.google.com](http://www.google.com)**

```
oot@balaji:/home/rachappa# sudo traceroute www.google.com
traceroute to www.google.com (142.250.195.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.632 ms  0.392 ms  0.229 ms

 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
```

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

```
oot@balaji:/home/rachappa# sudo traceroute www.google.com
traceroute to www.google.com (142.250.195.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.632 ms  0.392 ms  0.229 ms
```

**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](http://www.google.com)**



```

root@balaji:/home/rachappa# sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.195.164), 30 hops max, 60 byte packets
 1  10.0.2.2  0.583 ms  0.498 ms  0.453 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  *

```

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

**sudo traceroute -I [www.google.com](http://www.google.com)**

```

root@balaji:/home/rachappa# sudo traceroute -i www.google.com
Specify "host" missing argument.
root@balaji:/home/rachappa# sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.195.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.427 ms  0.188 ms  0.360 ms
 2  192.168.0.1 (192.168.0.1)  5.538 ms  5.331 ms  5.125 ms
 3  10.240.0.1 (10.240.0.1)  13.900 ms  13.689 ms  13.483 ms
 4  49.205.72.35.actcorp.in (49.205.72.35)  13.229 ms  12.974 ms  12.675 ms
 5  * * *
 6  * 10.248.5.23 (10.248.5.23)  7.553 ms  7.536 ms
 7  49.205.72.39.actcorp.in (49.205.72.39)  11.687 ms  8.581 ms  8.484 ms
 8  72.14.243.242 (72.14.243.242)  20.412 ms  19.974 ms  20.215 ms
 9  108.170.227.7 (108.170.227.7)  12.466 ms  12.174 ms  12.418 ms
10  142.251.55.91 (142.251.55.91)  11.365 ms  11.495 ms  12.214 ms
11  maa03s41-in-f4.1e100.net (142.250.195.164)  16.899 ms  11.622 ms  11.350 ms
root@balaji:/home/rachappa#

```

**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](http://www.google.com)**

```
root@balaji:/home/rachappa# sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.195.164), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.897 ms  0.840 ms  0.780 ms
 2 maa03s41-in-f4.1e100.net (142.250.195.164)  15.437 ms  16.198 ms  16.143 ms
root@balaji:/home/rachappa#
```

### **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](http://www.pes.edu)**

```
root@balaji:/home/rachappa# nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-27 23:28 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.0057s 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 5.09 seconds
root@balaji:/home/rachappa#
```

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

**nmap 163.53.78.128**

```

157/tcp open  https
Nmap done: 1 IP address (1 host up) scanned in 5.09 seconds
root@balaji:/home/rachappa# nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-27 23:29 IST
Nmap scan report for 163.53.78.128
Host is up (0.0035s 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 4.72 seconds
root@balaji:/home/rachappa# █

```

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

**nmap 192.168.1.1 192.168.1.2 192.168.1.3**

C

```

root@balaji:/home/rachappa# nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-27 23:31 IST
Nmap scan report for 192.168.1.1
Host is up (0.0011s latency).
All 1000 scanned ports on 192.168.1.1 are filtered

Nmap scan report for 192.168.1.2
Host is up (0.0016s latency).
All 1000 scanned ports on 192.168.1.2 are filtered

Nmap scan report for 192.168.1.3
Host is up (0.0015s latency).
All 1000 scanned ports on 192.168.1.3 are filtered

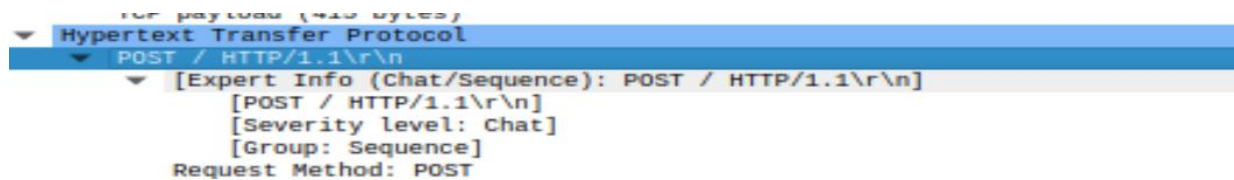
Nmap done: 3 IP addresses (3 hosts up) scanned in 28.74 seconds
root@balaji:/home/rachappa# █

```

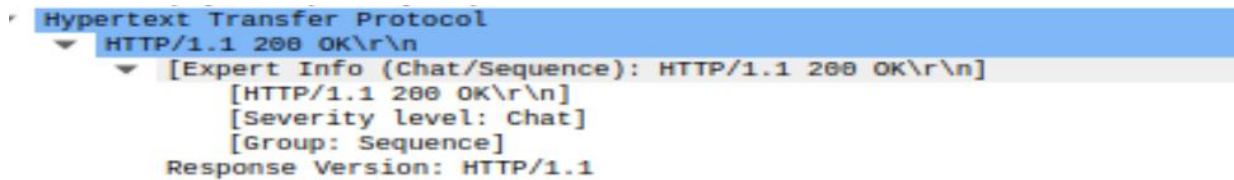
### Questions on above observations:

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

CLIENT



SERVER



SO they both run on http 1.1 version

2) When was the HTML file that you are retrieving last modified at the server?

5599	37.607886589	142.250.182.99	10.0.2.15	OCSP	758	Response
5685	40.118568204	10.0.2.15	142.250.182.99	OCSP	483	Request
5693	40.162576147	142.250.182.99	10.0.2.15	OCSP	758	Response
5840	45.477544100	10.0.2.15	34.107.221.82	HTTP	357	[TCP Previous segment not captured] GET /canonical
5846	45.492825323	34.107.221.82	10.0.2.15	HTTP	354	HTTP/1.1 200 OK (text/html)
5850	45.494645358	10.0.2.15	34.107.221.82	HTTP	359	[TCP Previous segment not captured] GET /success.t
5862	45.503999057	34.107.221.82	10.0.2.15	HTTP	272	HTTP/1.1 200 OK (text/plain)

3) How to tell ping to exit after a specified number of ECHO\_REQUEST packets?

ping -c 5 example.com

Where -c is use to count the number of ping

```
ping -c 5 example.com
```

4) How will you identify remote host apps and OS?

Using nmap