Week 1

NAME : SUNDEEP A	**SRN**: PES1UG20CS445
ROLL NO: 48	SECTION: H

Task 1: Linux Interface configuration(ifconfig/IP command)

Step 1: To display the status of all active network interfaces:

Interface name	IP address	MAC address
Ens33	IPv4 - 192.168.100.138 IPv6 - fe80::d1ff:29cb:a3b2:8064	00:0c:29:f9:4f:7d
lo	IPv4 – 127.0.0.1 IPv6 - ::1	00:0c:29:f9:4f:7d

```
pes1ug20cs445@sundeep:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.138  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::d1ff:29cb:a3b2:8064  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:f9:4f:7d  txqueuelen 1000  (Ethernet)
        RX packets 6783  bytes 9371223 (9.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1911  bytes 160922 (160.9 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 267  bytes 26141 (26.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 267  bytes 26141 (26.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2: To assign my IP address:

```
pes1ug20cs445@sundeep:~/Desktop$ sudo ifconfig ens33 10.0.8.48 netmask 255.255.255.0
pes1ug20cs445@sundeep:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.8.48  netmask 255.255.255.0  broadcast 10.0.8.255
        inet6 fe80::d1ff:29cb:a3b2:8064  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:f9:4f:7d  txqueuelen 1000  (Ethernet)
        RX packets 51351  bytes 69790443 (69.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15405  bytes 1132022 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Command used is – sudo ifconfig ens33 10.0.8.48 netmask 255.255.255.0
So, In the screenshot we can see that the IP address has been changed

Step 3: To activate and deactivate a network interface, type:
Deactivating ens33:

```
pes1ug20cs445@sundeep:~/Desktop$ sudo ifconfig ens33 down
pes1ug20cs445@sundeep:~/Desktop$ 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 1328  bytes 129887 (129.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1328  bytes 129887 (129.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pes1ug20cs445@sundeep:~/Desktop$
```

Thus, ens33 has been successfully deactivated

Activating ens33:

```
pes1ug20cs445@sundeep:~/Desktop$ sudo ifconfig ens33 up
pes1ug20cs445@sundeep:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.138  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::d1ff:29cb:a3b2:8064  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:f9:4f:7d  txqueuelen 1000  (Ethernet)
        RX packets 51650  bytes 69844246 (69.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15569  bytes 1153295 (1.1 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 1610  bytes 157125 (157.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1610  bytes 157125 (157.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pes1ug20cs445@sundeep:~/Desktop$
```

Step 4: To show the current neighbor

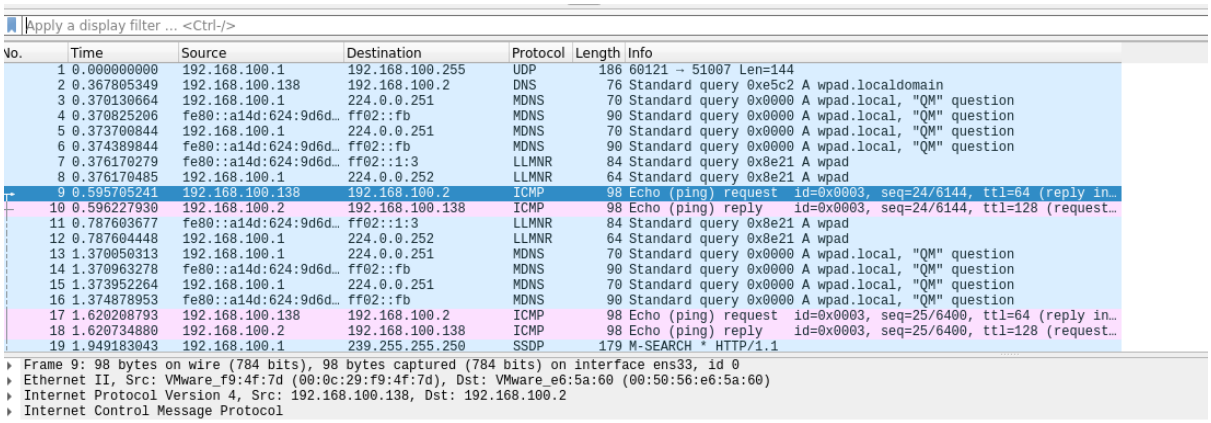
```
pes1ug20cs445@sundeep:~/Desktop$ ip neigh
192.168.100.2 dev ens33 lladdr 00:50:56:e6:5a:60 REACHABLE
pes1ug20cs445@sundeep:~/Desktop$
```

Task 2:Ping PDU Capture

Step 1:Assign an IP address to the system.

```
pes1ug20cs445@sundeep:~/Desktop$ sudo ifconfig ens33 10.0.8.48 netmask 255.255.255.0
pes1ug20cs445@sundeep:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
       inet 10.0.8.48  netmask 255.255.255.0  broadcast 10.0.8.255
       inet6 fe80::d1ff:29cb:a3b2:8064  prefixlen 64  scopeid 0x20<link>
       ether 00:0c:29:f9:4f:7d  txqueuelen 1000  (Ethernet)
       RX packets 51351  bytes 69790443 (69.7 MB)
       RX errors 0  dropped 0  overruns 0  frame 0
       TX packets 15405  bytes 1132022 (1.1 MB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2: Launch Wireshark and select ‘any’ interface



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

```
pes1ug20cs445@sundeep:~/Desktop$ ping 10.0.8.48
PING 10.0.8.48 (10.0.8.48) 56(84) bytes of data.
64 bytes from 10.0.8.48: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 10.0.8.48: icmp_seq=2 ttl=64 time=0.105 ms
64 bytes from 10.0.8.48: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.8.48: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 10.0.8.48: icmp_seq=5 ttl=64 time=0.036 ms
64 bytes from 10.0.8.48: icmp_seq=6 ttl=64 time=0.032 ms
64 bytes from 10.0.8.48: icmp_seq=7 ttl=64 time=0.094 ms
64 bytes from 10.0.8.48: icmp_seq=8 ttl=64 time=0.117 ms
64 bytes from 10.0.8.48: icmp_seq=9 ttl=64 time=0.048 ms
64 bytes from 10.0.8.48: icmp_seq=10 ttl=64 time=0.086 ms
64 bytes from 10.0.8.48: icmp_seq=11 ttl=64 time=0.085 ms
64 bytes from 10.0.8.48: icmp_seq=12 ttl=64 time=0.053 ms
64 bytes from 10.0.8.48: icmp_seq=13 ttl=64 time=0.052 ms
64 bytes from 10.0.8.48: icmp_seq=14 ttl=64 time=0.080 ms
64 bytes from 10.0.8.48: icmp_seq=15 ttl=64 time=0.150 ms
^C
--- 10.0.8.48 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14337ms
rtt min/avg/max/mdev = 0.031/0.073/0.150/0.035 ms
pes1ug20cs445@sundeep:~/Desktop$
```

Step 4: Observations to be made

TTL	64
Protocol Used By Ping	ICMP
Time	Order of 10^{-2}

Step 5: Analyze the following in Wireshark
I am pinging my neighbor's system
HTTP Request:

HTTP Response:

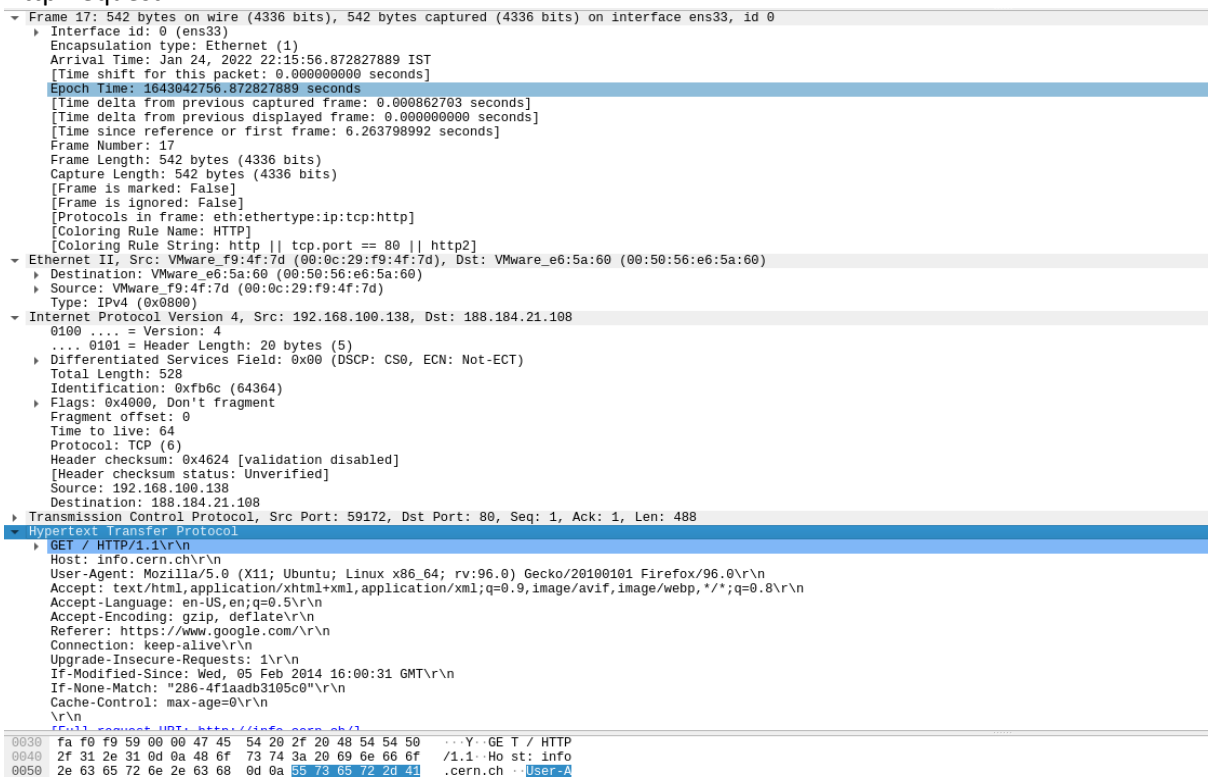
Details	First Request	First Reply
Frame Number	9	10
Source IP address	192.168.100.138	192.168.100.2
Destination IP address	192.168.100.2	192.168.100.138
ICMP Type Value	8	0

ICMP Code Value	0	0
Source Ethernet Address	00:0c:29:f9:4f:7d	00:50:56:e6:5a:60
Destination Ethernet Address	00:50:56:e6:5a:60	00:0c:29:f9:4f:7d
Internet Protocol Version	4	4
Time To Live Value	64	128

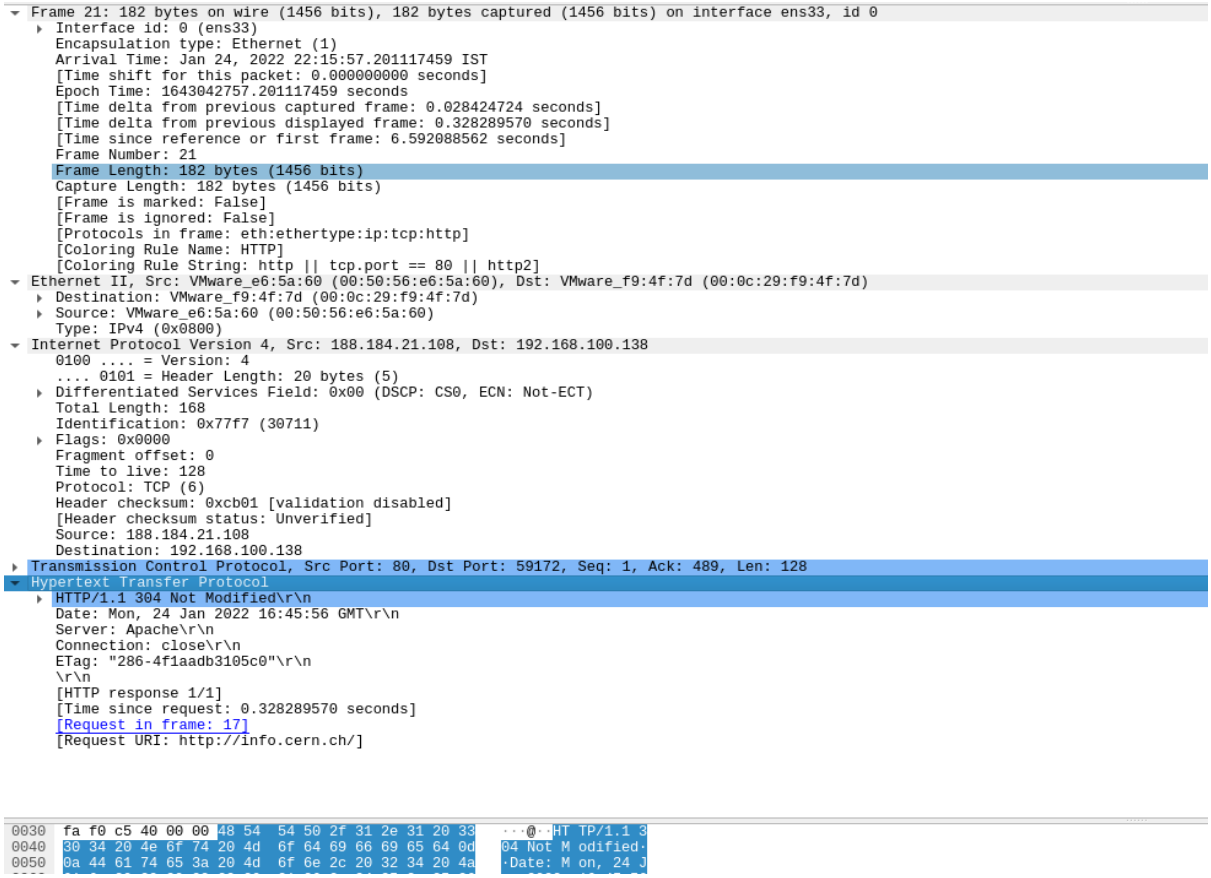
Task 3: HTTP PDU Capture

Since Flipkart.com is a https protocol , I have used info.cern.ch website which has http protocol.

Http Request:



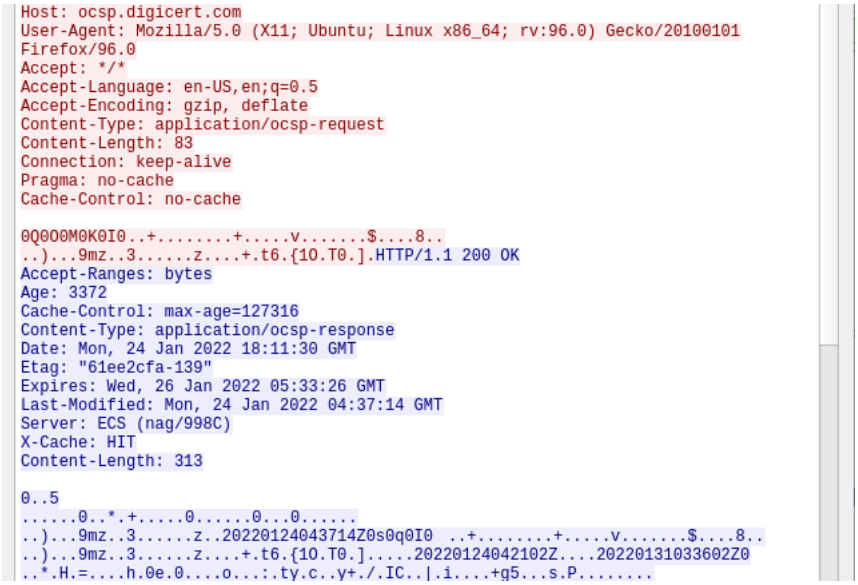
Http response:



Step 3: Analyze the first (interaction of host to the web server) and second frame (response of server to the client).

Details	First Echo Request	First Echo Response
Frame Number	17	21
Source IP address	192.168.100.138	188.184.21.108
Destination IP address	188.184.21.108	192.168.100.138
Source Port	59172	80
Destination Port	80	59172
Source Ethernet Address	00:0c:29:f9:f4:7d	00:50:56:e6:5a:60
Destination Ethernet Address	00:50:56:e6:5a:60	00:0c:29:f9:f4:7d

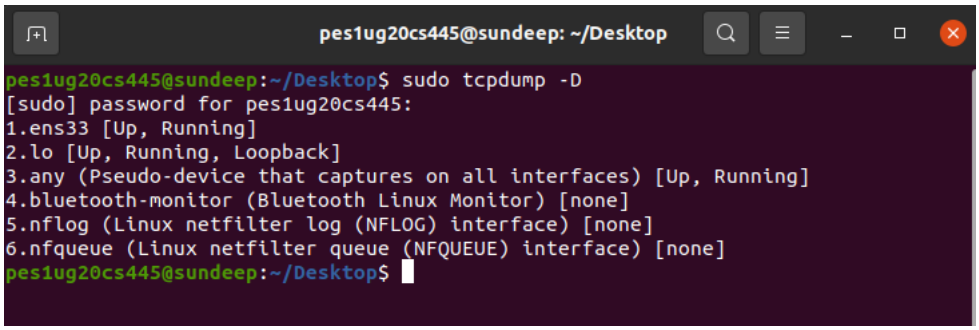
Step 4: Analyze the HTTP request and response and complete the table.



HTTP Request		HTTP Response	
Get	GET / HTTP/1.1	Server	ECS(nag/998C)
Host	ocsp.digicert.com	Content-Type	Application/ocsp-response
Accept-Language	En-US	Date	24 jan 2022 5:33:26 GMT
Accept-Encoding	gzip,deflate	Location	ocsp.digicert.com
Connection	Keep-alive	Connection	Keep-alive
User-agent	Mozilla/5.0 Gecko/20100101 Firefox/96.0	Content length	313

Task 4: Capturing Packets with TCP dump

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



Step 2: Capture all packets in any interface

```
pes1ug20cs445@sundeep:~/Desktop$ sudo tcpdump -i any
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
23:58:00.187745 IP sundeep > bom07s24-in-f4.1e100.net: ICMP echo request, id 1, seq 19, length 64
23:58:00.189699 IP localhost.41506 > localhost.domain: 12249+ [1au] PTR? 228.67.250.142.in-addr.arpa. (56)
23:58:00.190430 IP sundeep.60834 > _gateway.domain: 64120+ PTR? 228.67.250.142.in-addr.arpa. (45)
23:58:00.224504 IP bom07s24-in-f4.1e100.net > sundeep: ICMP echo reply, id 1, seq 19, length 64
23:58:00.226969 IP _gateway.domain > sundeep.60834: 64120 1/0/0 PTR bom07s24-in-f4.1e100.net. (83)
23:58:00.227486 IP localhost.domain > localhost.41506: 12249 1/0/1 PTR bom07s24-in-f4.1e100.net. (94)
23:58:00.265143 IP localhost.59583 > localhost.domain: 25953+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
23:58:01.190483 IP sundeep > bom07s24-in-f4.1e100.net: ICMP echo request, id 1, seq 20, length 64
23:58:01.225410 IP bom07s24-in-f4.1e100.net > sundeep: ICMP echo reply, id 1, seq 20, length 64
23:58:01.534913 IP 192.168.100.1.60856 > 192.168.100.255.51007: UDP, length 144
23:58:01.535399 IP localhost.36949 > localhost.domain: 56936+ [1au] PTR? 255.100.168.192.in-addr.arpa. (57)
23:58:01.536097 IP sundeep.52425 > _gateway.domain: 53906+ PTR? 255.100.168.192.in-addr.arpa. (46)
23:58:01.570889 IP _gateway.domain > sundeep.52425: 53906 NXDomain 0/1/0 (95)
23:58:01.571599 IP localhost.domain > localhost.36949: 56936 NXDomain 0/0/1 (57)
^C
14 packets captured
39 packets received by filter
18 packets dropped by kernel
pes1ug20cs445@sundeep:~/Desktop$
```

Step 3: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets

```
pes1ug20cs445@sundeep:~/Desktop$ sudo tcpdump -i any -c5 icmp
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
23:59:56.190219 IP sundeep > bom07s24-in-f4.1e100.net: ICMP echo request, id 2, seq 6, length 64
23:59:56.227229 IP bom07s24-in-f4.1e100.net > sundeep: ICMP echo reply, id 2, seq 6, length 64
23:59:57.192436 IP sundeep > bom07s24-in-f4.1e100.net: ICMP echo request, id 2, seq 7, length 64
23:59:57.229243 IP bom07s24-in-f4.1e100.net > sundeep: ICMP echo reply, id 2, seq 7, length 64
23:59:58.194246 IP sundeep > bom07s24-in-f4.1e100.net: ICMP echo request, id 2, seq 8, length 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
pes1ug20cs445@sundeep:~/Desktop$
```

Step 4: : Check the packet content. For example, inspect the HTTP content of a web request

```
pes1ug20cs445@sundeep:~/Desktop$ 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
00:01:57.958225 IP 192.168.100.138.45396 > 35.232.111.17.80: Flags [S], seq 886004043, win 64240, options [mss 1460,sackOK,TS val 137418184 ecr 0,nop,wscale 7], length 0
E..<A.@.@.A...d.#.o..T.P4.YK.........Z.........
.0..........
00:01:58.974062 IP 192.168.100.138.45396 > 35.232.111.17.80: Flags [S], seq 886004043, win 64240, options [mss 1460,sackOK,TS val 137419200 ecr 0,nop,wscale 7], length 0
E..<A.@.@.A...d.#.o..T.P4.YK.........Z.........
.0..........
00:01:59.250099 IP 35.232.111.17.80 > 192.168.100.138.45396: Flags [S.], seq 1259145604, ack 886004044, win 64240, options [mss 1460], length 0
E..,..t.....+#.o...d..P.TK.      .4.YL`...P.........
00:01:59.250259 IP 192.168.100.138.45396 > 35.232.111.17.80: Flags [.], ack 1, win 64240, length 0
E..(A.@.@.A...d.#.o..T.P4.YLK.  .P....F..
00:01:59.250692 IP 192.168.100.138.45396 > 35.232.111.17.80: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HTTP: GET / HTTP/1.1
E...A.@.@.AJ..d.#.o..T.P4.YLK.  .P.......GET / HTTP/1.1
Host: connectivity-check.ubuntu.com
Accept: */*
Connection: close


00:01:59.251412 IP 35.232.111.17.80 > 192.168.100.138.45396: Flags [.], ack 88, win 64240, length 0
E..(.u......#.o...d..P.TK.      .4.Y.P...h.........
00:01:59.637841 IP 35.232.111.17.80 > 192.168.100.138.45396: Flags [FP.], seq 1:149, ack 88, win 64240, length 148: HTTP: HTTP/1.1 204 No Content
E....v......#.o...d..P.TK.      .4.Y.P....Q..HTTP/1.1 204 No Content
Date: Mon, 24 Jan 2022 18:31:59 GMT
Server: Apache/2.4.18 (Ubuntu)
X-NetworkManager-Status: online
Connection: close


00:01:59.638322 IP 192.168.100.138.45396 > 35.232.111.17.80: Flags [F.], seq 88, ack 150, win 64091, length 0
E..(A.@.@.A...d.#.o..T.P4.Y.K
.P..[.F..
00:01:59.638951 IP 35.232.111.17.80 > 192.168.100.138.45396: Flags [.], ack 89, win 64239, length 0
E..(.w.....,#.o...d..P.TK
.4.Y.P...gy.......
00:02:03.845642 IP 192.168.100.138.34676 > 216.92.49.183.80: Flags [S], seq 3192748716, win 64240, options [mss 1460,sackOK,TS val 1150092012 ecr 0,nop,wscale 7], length 0
E..<..@.@.....d..\1..t.P.Mz........./u.........
D...........
10 packets captured
10 packets received by filter
0 packets dropped by kernel
pes1ug20cs445@sundeep:~/Desktop$
```

Step 5: To save packets to a file instead of displaying them on screen

```
pes1ug20cs445@sundeep:~/Desktop$ 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
28 packets received by filter
0 packets dropped by kernel
pes1ug20cs445@sundeep:~/Desktop$
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute

```
pes1ug20cs445@sundeep:~/Desktop$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.67.228), 30 hops max, 60 byte packets
 1  _gateway (192.168.100.2)  1.749 ms  0.637 ms  0.204 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
pes1ug20cs445@sundeep:~/Desktop$
```

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

Destination Address – 142.25067.228

Max number of hops – 30

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames

```
pes1ug20cs445@sundeep:~/Desktop$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.67.228), 30 hops max, 60 byte packets
 1  192.168.100.2  0.352 ms  0.231 ms  0.161 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
pes1ug20cs445@sundeep:~/Desktop$
```

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

```
pes1ug20cs445@sundeep:~/Desktop$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.67.228), 30 hops max, 60 byte packets
 1  _gateway (192.168.100.2)  0.682 ms  0.578 ms  0.481 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  bom07s24-in-f4.1e100.net (142.250.67.228)  34.345 ms  35.021 ms  39.275 ms
pes1ug20cs445@sundeep:~/Desktop$
```

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

```
pes1ug20cs445@sundeep:~/Desktop$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.67.228), 30 hops max, 60 byte packets
 1  _gateway (192.168.100.2)  1.435 ms  0.397 ms  0.199 ms
 2  bom07s24-in-f4.1e100.net (142.250.67.228)  33.666 ms  37.041 ms  36.983 ms
pes1ug20cs445@sundeep:~/Desktop$
```

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.

```
pes1ug20cs445@sundeep:~/Desktop$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-25 00:18 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.042s 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 93.03 seconds
```

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

```
pes1ug20cs445@sundeep:~/Desktop$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-25 00:22 IST
Nmap scan report for 163.53.78.128
Host is up (0.057s 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 69.14 seconds
```

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

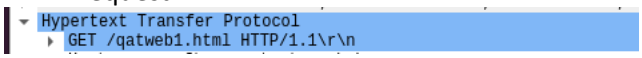
```
pes1ug20cs445@sundeep:~/Desktop$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-25 00:24 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.09 seconds
pes1ug20cs445@sundeep:~/Desktop$
```

Questions on above observations:

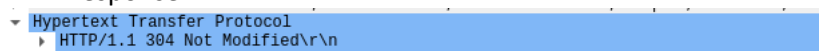
1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

Answer - The Firefox browser used is running HTTP vl.1, and this can be seen in the request header which contains the method (GET) followed by the HTTP version. Similarly, the HTTP version of the web server is v1.1 and can be seen in the header of the HTTP response sent back to the browser.

HTTP Request:



HTTP response:



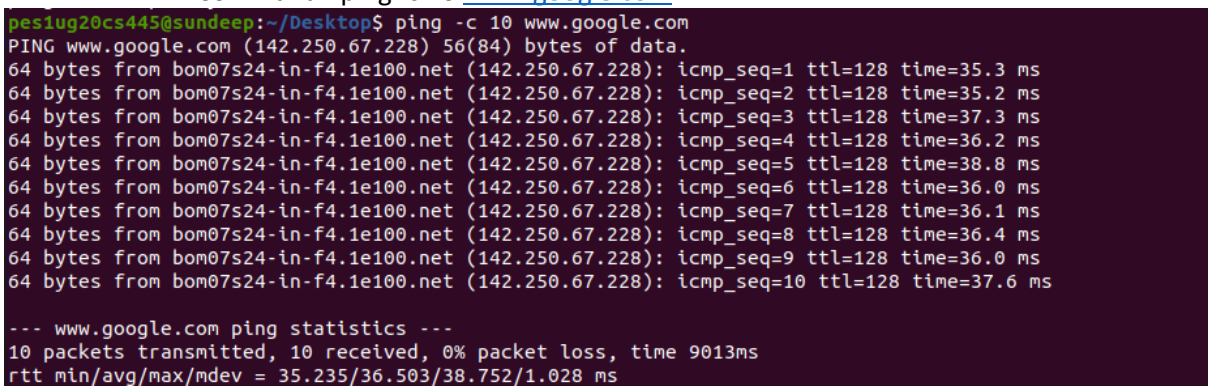
2.When was the HTML file that you are retrieving last modified at the server?

Answer - We can find the last modified time of the HTML file at the server by observing the Last-Modified field of the HTTP response object. The Last-Modified field stores a timestamp of the last modification time.

3.How to tell ping to exit after a specified number of ECHO_REQUEST packets?

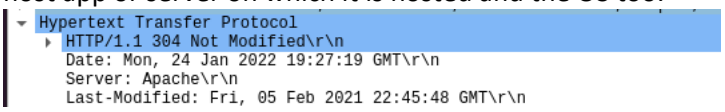
Answer - Ping continues to send ICMP packages until it receives an interrupt signal. To specify the number of ECH_REQUEST packages after which ping will exit, we can use the - c option followed by the number of packages.

Command: ping -c 10 www.google.com



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

Answer: 1. We can obtain the remote host app and OS of the server by observing the Server files of the HTTP response object. The Server field stores the remote host app or server on which it is hosted and the OS too.



2. We can use nmap to find the OS too. It will scan the network to find information about the remote host apps and OS.