NAME: H M Mythreya	SRN:PES2UG20CS130	SECTION:C
	25/1/22	WEEK:1

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

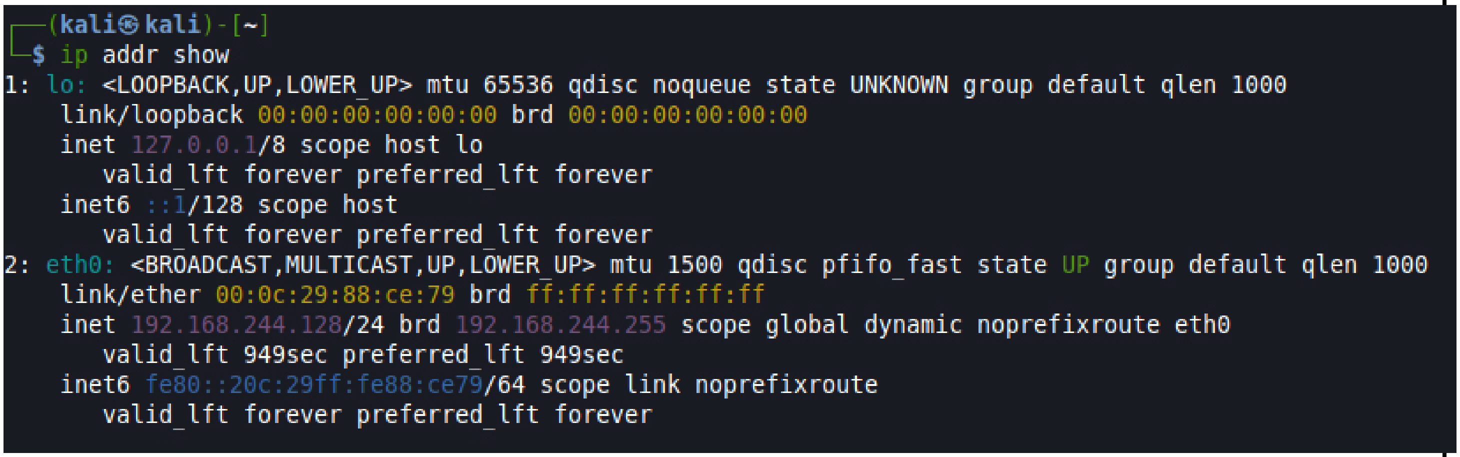
IMPORTANT INSTRUCTIONS:

- 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 to Edmodo as a single PDF file. (Refer general guidelines for submission requirements).
- 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**



```
┌──(kali㉿kali)-[~]
└─$ 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:88:ce:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.244.128/24 brd 192.168.244.255 scope global dynamic noprefixroute eth0
       valid_lft 949sec preferred_lft 949sec
    inet6 fe80::20c:29ff:fe88:ce79/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Analye and fill the following table:

ip address table:

Interface name	**IP address (IPv4 / IPv6)**	**MAC address**
eth0	Ipv4:192.168.244.128 Ipv6: fe80::20c:29ff:fe88:ce79	00:0c:29:88:ce:79
lo	Ipv4: 127.0.0.1 Ipv6: ::1/128	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)

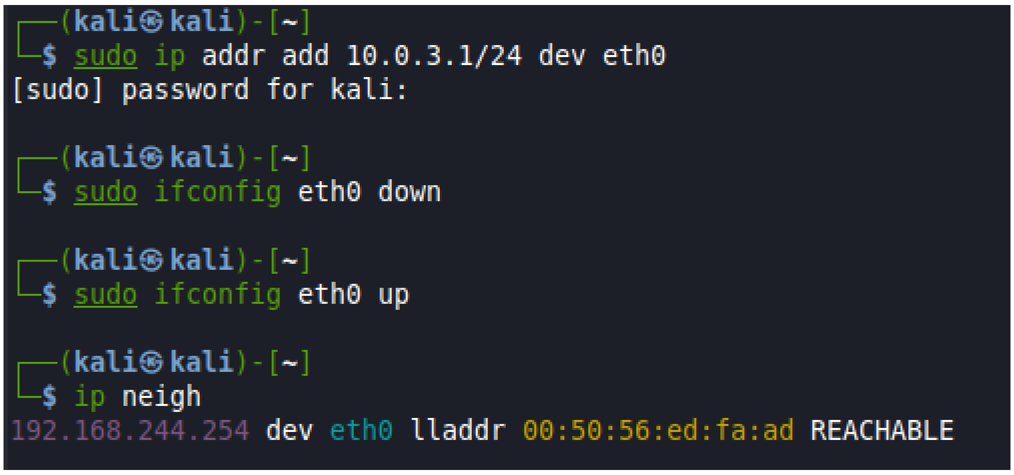
sudo ip addr add 10.0.your_section.your_sno /24 dev interface_name

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**

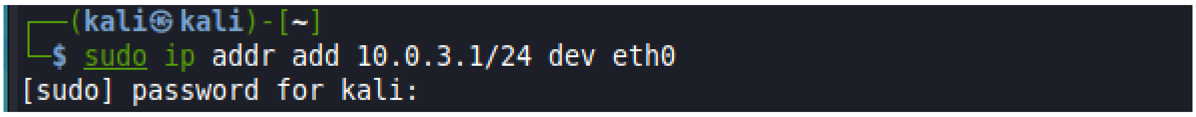


```
┌──(kali㉿kali)-[~]
└─$ sudo ip addr add 10.0.3.1/24 dev eth0
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo ifconfig eth0 down

┌──(kali㉿kali)-[~]
└─$ sudo ifconfig eth0 up

┌──(kali㉿kali)-[~]
└─$ ip neigh
192.168.244.254 dev eth0 lladdr 00:50:56:ed:fa:ad REACHABLE
```

Task 2: Ping PDU (Packet Data Units or Packets) Capture

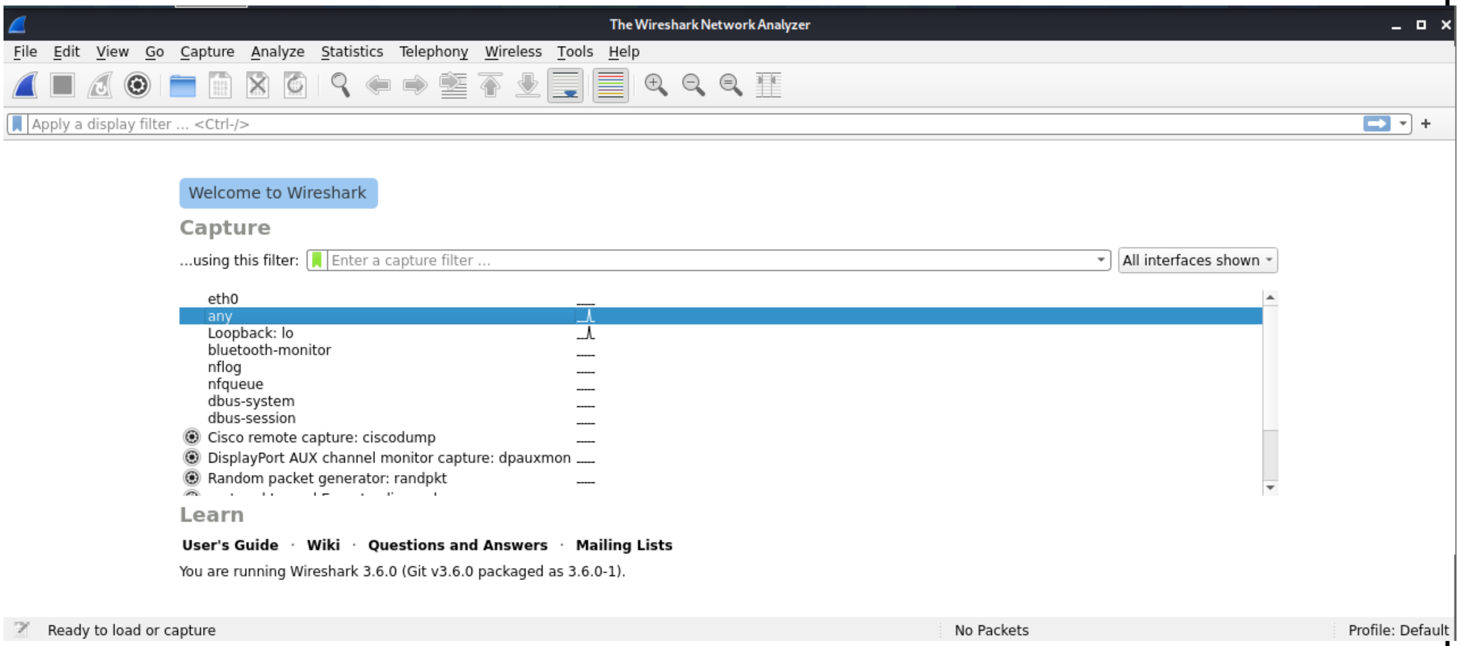
Step 1: Assign an IP address to the system (Host).

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



```
┌──(kali㉿kali)-[~]
└─$ sudo ip addr add 10.0.3.1/24 dev eth0
[sudo] password for kali:
```

Step 2: Launch Wireshark and select 'any' interface



The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter … <Ctrl-/>

Welcome to Wireshark

Capture

…using this filter: Enter a capture filter … All interfaces shown

eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt

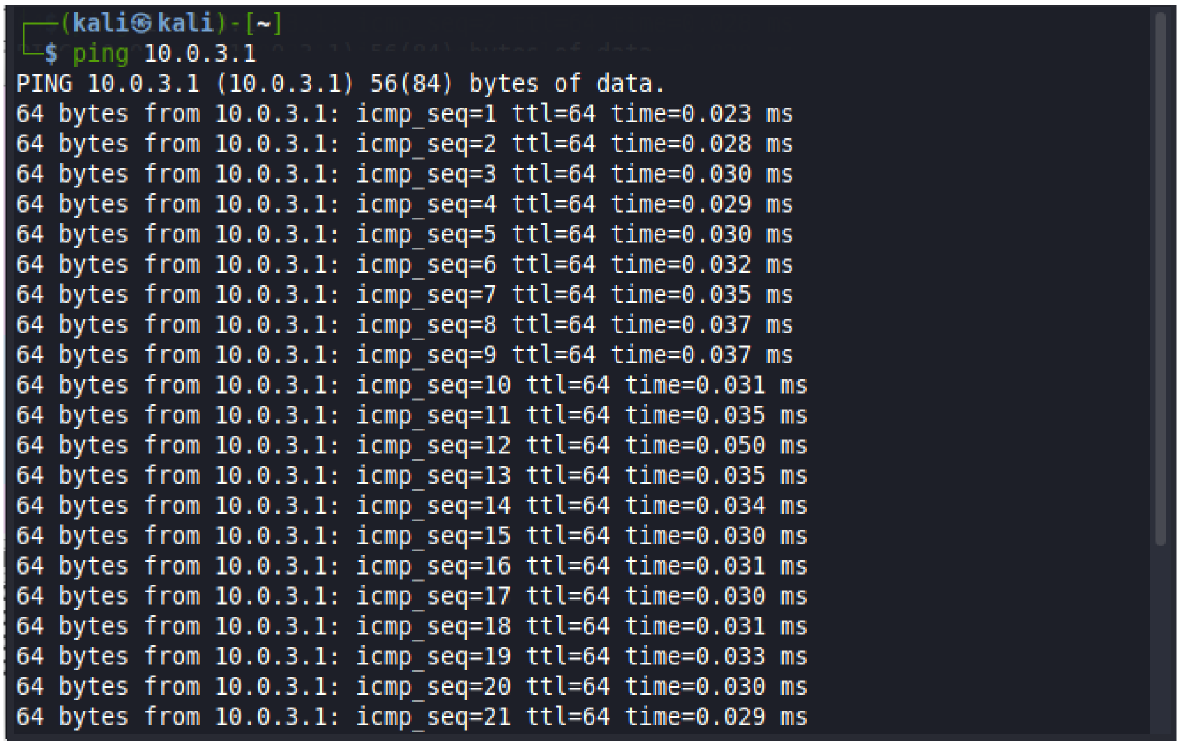
Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.6.0 (Git v3.6.0 packaged as 3.6.0-1).

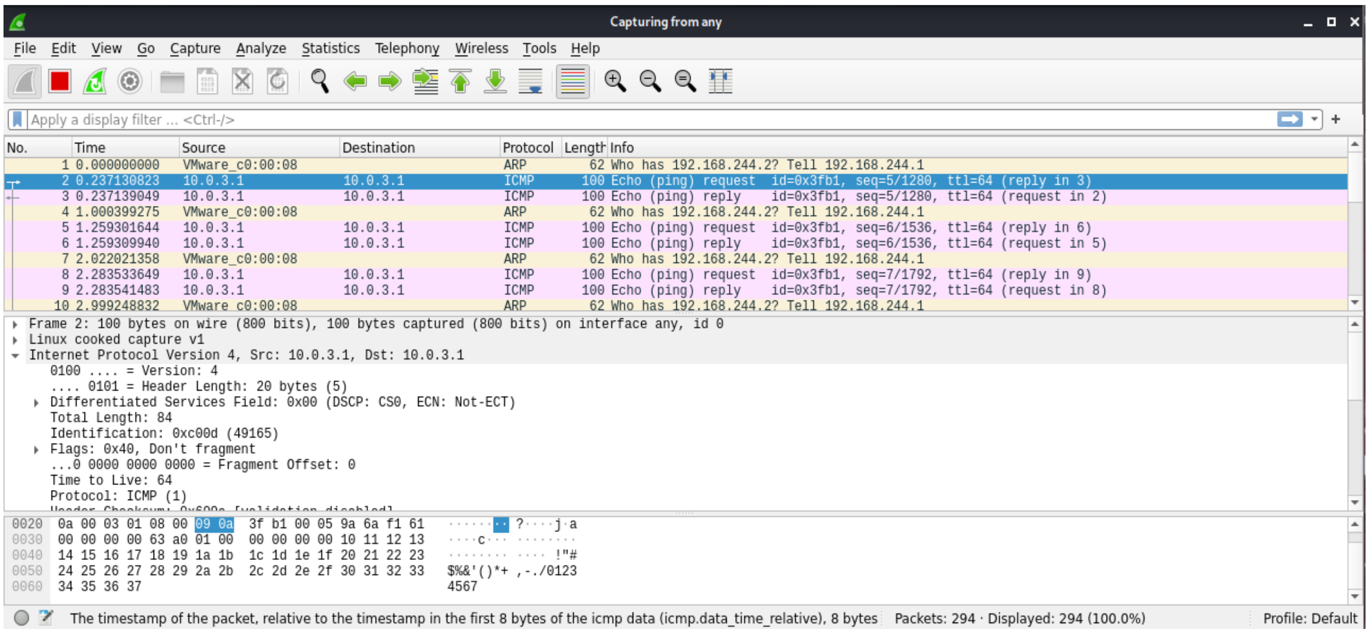
Ready to load or capture No Packets Profile: Default

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



```
┌──(kali㉿kali)-[~]
└─$ ping 10.0.3.1
PING 10.0.3.1 (10.0.3.1) 56(84) bytes of data.
64 bytes from 10.0.3.1: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from 10.0.3.1: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from 10.0.3.1: icmp_seq=5 ttl=64 time=0.030 ms
64 bytes from 10.0.3.1: icmp_seq=6 ttl=64 time=0.032 ms
64 bytes from 10.0.3.1: icmp_seq=7 ttl=64 time=0.035 ms
64 bytes from 10.0.3.1: icmp_seq=8 ttl=64 time=0.037 ms
64 bytes from 10.0.3.1: icmp_seq=9 ttl=64 time=0.037 ms
64 bytes from 10.0.3.1: icmp_seq=10 ttl=64 time=0.031 ms
64 bytes from 10.0.3.1: icmp_seq=11 ttl=64 time=0.035 ms
64 bytes from 10.0.3.1: icmp_seq=12 ttl=64 time=0.050 ms
64 bytes from 10.0.3.1: icmp_seq=13 ttl=64 time=0.035 ms
64 bytes from 10.0.3.1: icmp_seq=14 ttl=64 time=0.034 ms
64 bytes from 10.0.3.1: icmp_seq=15 ttl=64 time=0.030 ms
64 bytes from 10.0.3.1: icmp_seq=16 ttl=64 time=0.031 ms
64 bytes from 10.0.3.1: icmp_seq=17 ttl=64 time=0.030 ms
64 bytes from 10.0.3.1: icmp_seq=18 ttl=64 time=0.031 ms
64 bytes from 10.0.3.1: icmp_seq=19 ttl=64 time=0.033 ms
64 bytes from 10.0.3.1: icmp_seq=20 ttl=64 time=0.030 ms
64 bytes from 10.0.3.1: icmp_seq=21 ttl=64 time=0.029 ms
```

Observations to be made



Capturing from any
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   VMware_c0:00:08                  ARP       62 Who has 192.168.244.2? Tell 192.168.244.1
    2 0.237130823   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) request  id=0x3fb1, seq=5/1280, ttl=64 (reply in 3)
    3 0.237139049   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) reply    id=0x3fb1, seq=5/1280, ttl=64 (request in 2)
    4 1.000399275   VMware_c0:00:08                  ARP       62 Who has 192.168.244.2? Tell 192.168.244.1
    5 1.259301644   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) request  id=0x3fb1, seq=6/1536, ttl=64 (reply in 6)
    6 1.259309940   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) reply    id=0x3fb1, seq=6/1536, ttl=64 (request in 5)
    7 2.022021358   VMware_c0:00:08                  ARP       62 Who has 192.168.244.2? Tell 192.168.244.1
    8 2.283533649   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) request  id=0x3fb1, seq=7/1792, ttl=64 (reply in 9)
    9 2.283541483   10.0.3.1         10.0.3.1        ICMP     100 Echo (ping) reply    id=0x3fb1, seq=7/1792, ttl=64 (request in 8)
   10 2.999248832   VMware_c0:00:08                  ARP       62 Who has 192.168.244.2? Tell 192.168.244.1

▸ 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: 10.0.3.1, Dst: 10.0.3.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xc00d (49165)
  ▸ Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)

0020  0a 00 03 01 08 00 09 0a  3f b1 00 05 9a 6a f1 61   ········?····j·a
0030  00 00 00 00 63 a0 01 00  00 00 00 00 10 11 12 13   ····c···········
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

The timestamp of the packet, relative to the timestamp in the first 8 bytes of the icmp data (icmp.data_time_relative), 8 bytes   Packets: 294 · Displayed: 294 (100.0%)   Profile: Default
```

Step 4: Analyze the following in Terminal

- TTL: 64
- Protocol used by ping: ICMP
- Time: 10661ms

Step 5: Analyze the following in Wireshark

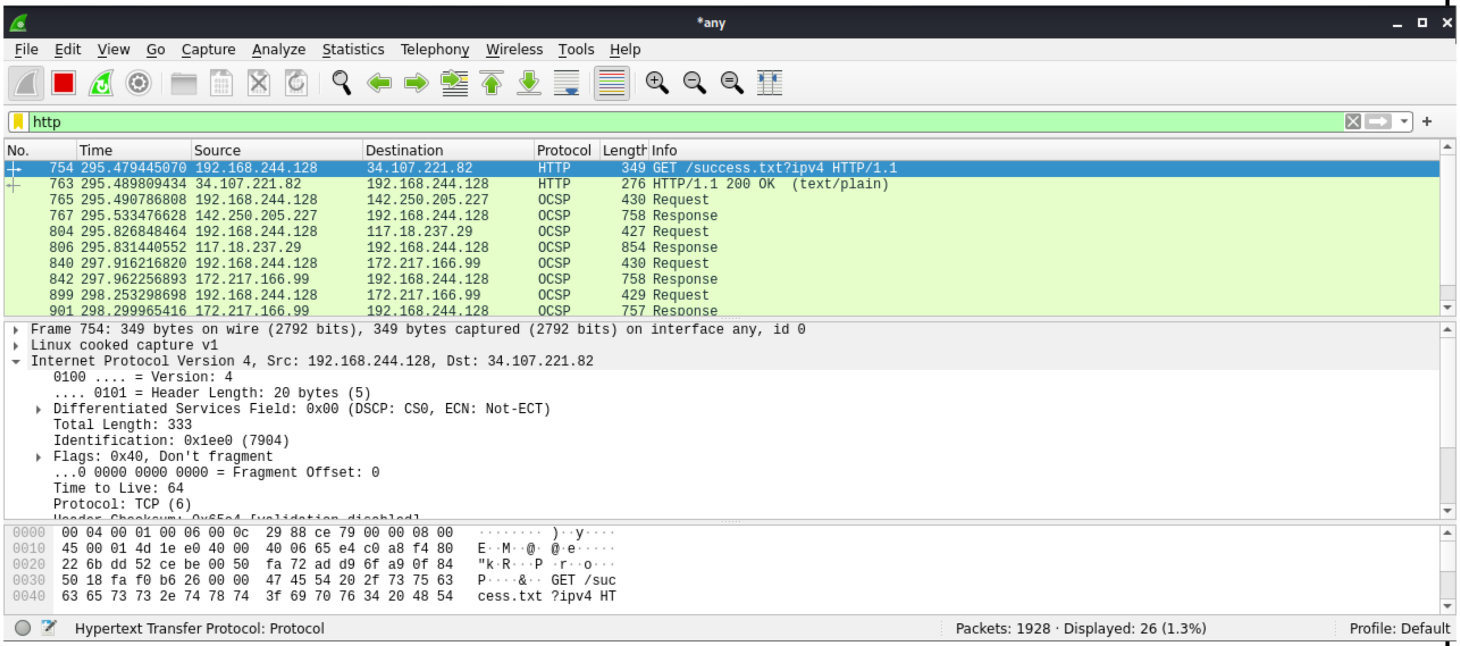
On Packet List Pane, select the first echo packet on the list. On Packet Details Pane, click on each of the four "+" to expand the information. Analyze the frames with the first echo request and echo reply and complete the table below.

Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.3.1	10.0.3.1
Destination IP address	10.0.3.1	10.0.3.1
ICMP Type Value	1	1
ICMP Code Value	1	1
Source Ethernet Address	**00:0c:29:88:ce:79**	**00:0c:29:88:ce:79**
Destination Ethernet Address	**00:0c:29:88:ce:79**	**00:0c:29:88:ce:79**
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


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

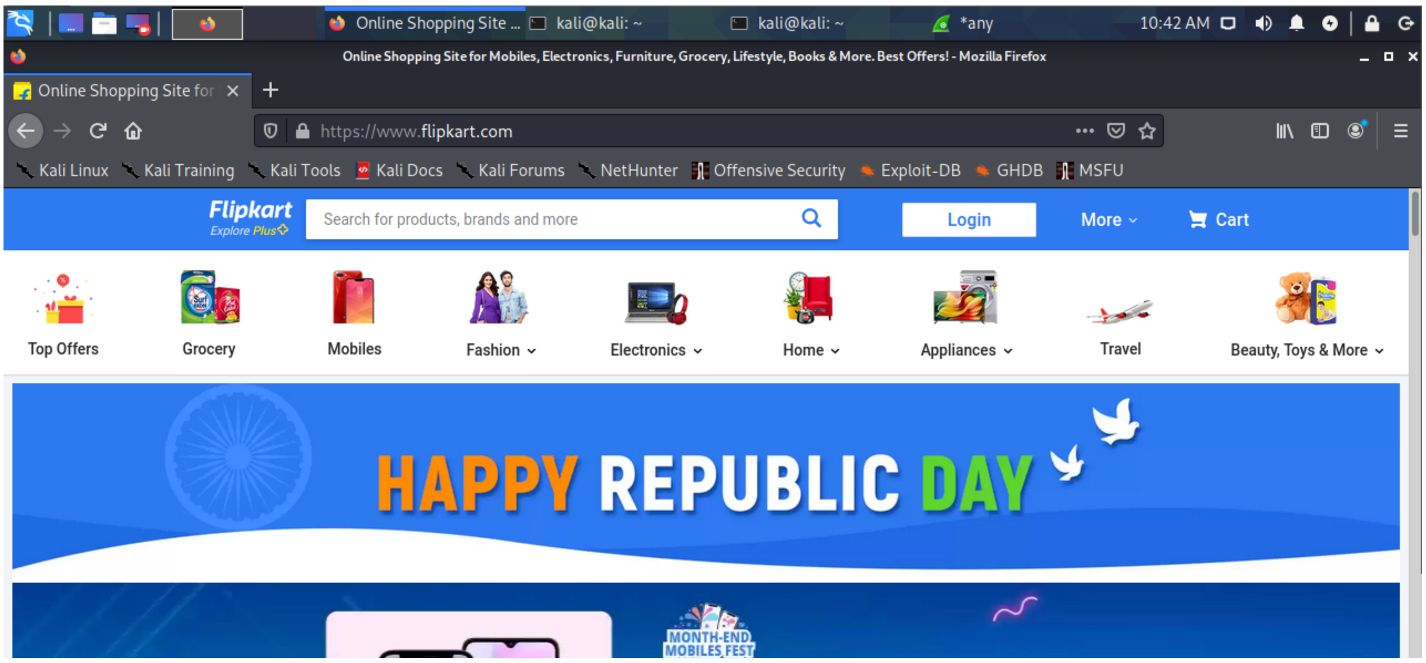
```
No.   Time           Source           Destination      Protocol Length Info
  754 295.479445070 192.168.244.128  34.107.221.82    HTTP     349 GET /success.txt?ipv4 HTTP/1.1
  763 295.489809434 34.107.221.82    192.168.244.128  HTTP     276 HTTP/1.1 200 OK  (text/plain)
  765 295.490786808 192.168.244.128  142.250.205.227  OCSP     430 Request
  767 295.533476628 142.250.205.227  192.168.244.128  OCSP     758 Response
  804 295.826848464 192.168.244.128  117.18.237.29    OCSP     427 Request
  806 295.831440552 117.18.237.29    192.168.244.128  OCSP     854 Response
  840 297.916216820 192.168.244.128  172.217.166.99   OCSP     430 Request
  842 297.962256893 172.217.166.99   192.168.244.128  OCSP     758 Response
  899 298.253298698 192.168.244.128  172.217.166.99   OCSP     429 Request
  901 298.299965416 172.217.166.99   192.168.244.128  OCSP     757 Response
```

```
▸ Frame 754: 349 bytes on wire (2792 bits), 349 bytes captured (2792 bits) on interface any, id 0
▸ Linux cooked capture v1
▾ Internet Protocol Version 4, Src: 192.168.244.128, Dst: 34.107.221.82
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 333
    Identification: 0x1ee0 (7904)
  ▸ Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
```

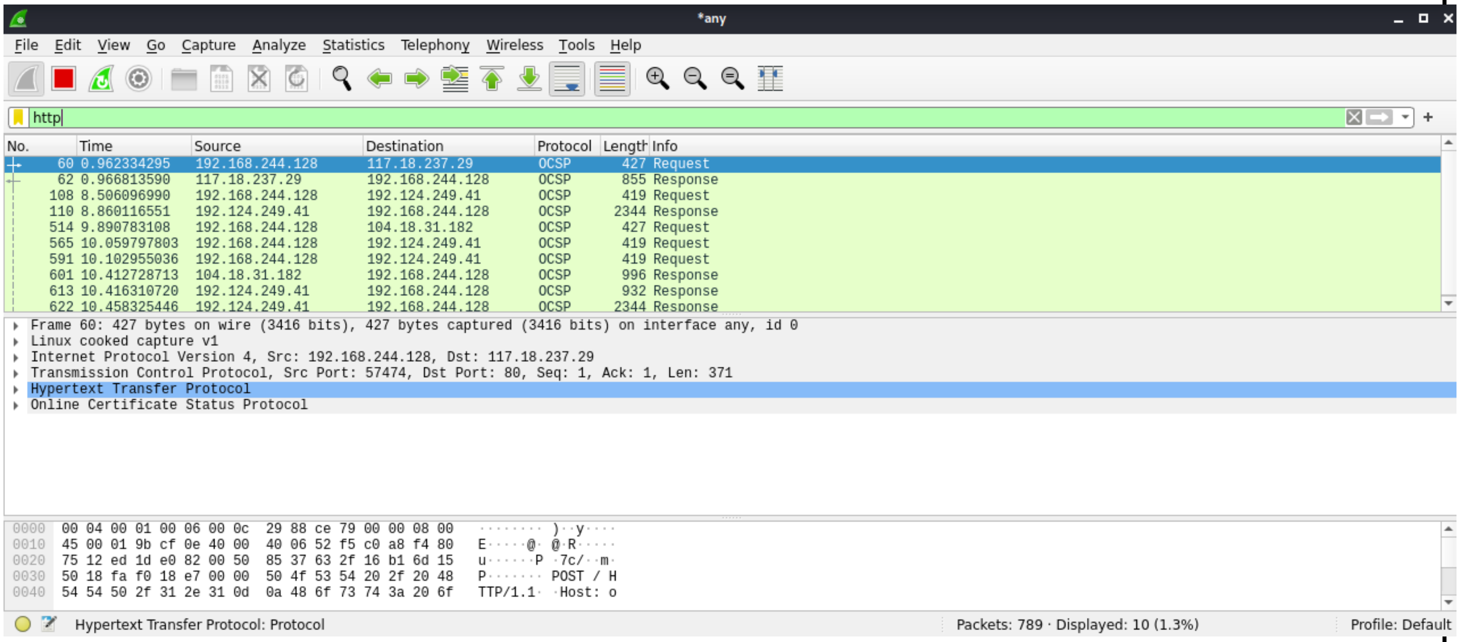
```
0000  00 04 00 01 00 06 00 0c  29 88 ce 79 00 00 08 00   ········ )··y····
0010  45 00 01 4d 1e e0 40 00  40 06 65 e4 c0 a8 f4 80   E··M··@· @·e·····
0020  22 6b dd 52 ce be 00 50  fa 72 ad d9 6f a9 0f 84   "k·R···P ·r··o···
0030  50 18 fa f0 b6 26 00 00  47 45 54 20 2f 73 75 63   P····&·· GET /suc
0040  63 65 73 73 2e 74 78 74  3f 69 70 76 34 20 48 54   cess.txt ?ipv4 HT
```

Hypertext Transfer Protocol: Protocol   Packets: 1928 · Displayed: 26 (1.3%)   Profile: Default

Step 2: Open Firefox browser, and browse www.flipkart.com



Observations to be made



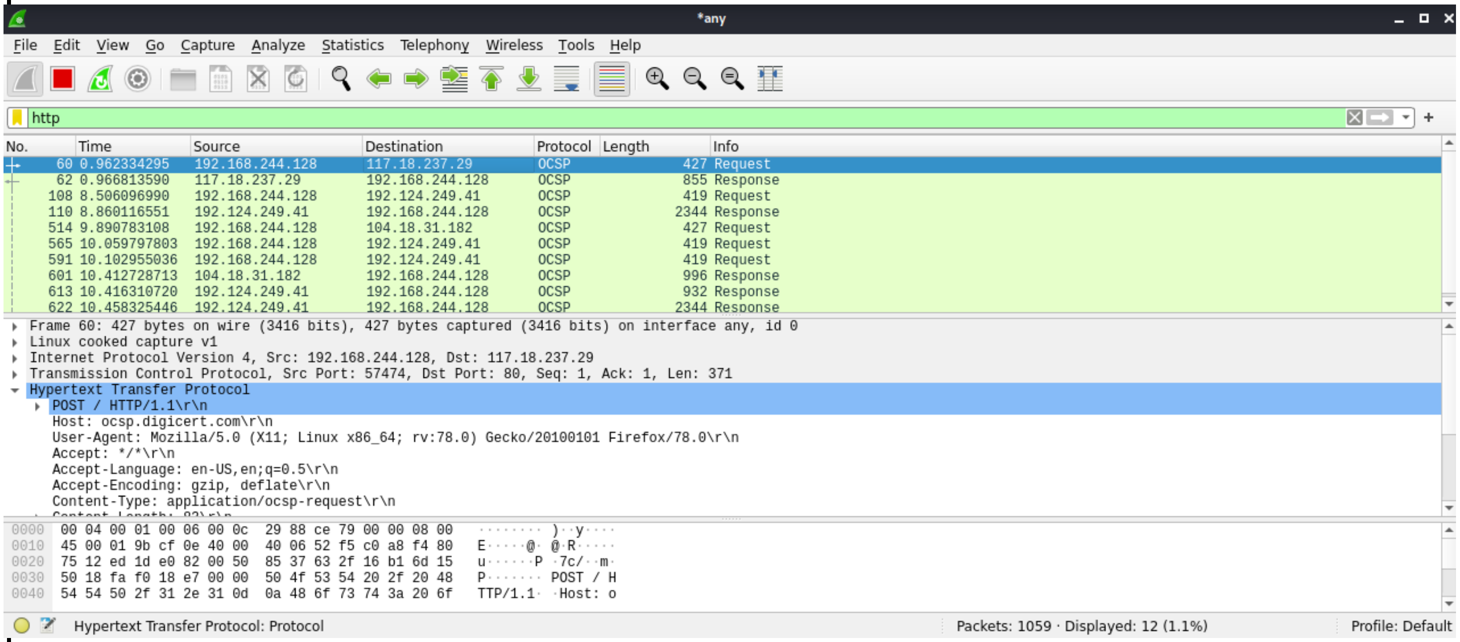
```
http
```

```
No.     Time          Source            Destination       Protocol  Length Info
    60 0.962334295   192.168.244.128   117.18.237.29     OCSP       427 Request
    62 0.966813590   117.18.237.29     192.168.244.128   OCSP       855 Response
   108 8.506096990   192.168.244.128   192.124.249.41    OCSP       419 Request
   110 8.860116551   192.124.249.41    192.168.244.128   OCSP      2344 Response
   514 9.890783108   192.168.244.128   104.18.31.182     OCSP       427 Request
   565 10.059797803  192.168.244.128   192.124.249.41    OCSP       419 Request
   591 10.102955036  192.168.244.128   192.124.249.41    OCSP       419 Request
   601 10.412728713  104.18.31.182     192.168.244.128   OCSP       996 Response
   613 10.416310720  192.124.249.41    192.168.244.128   OCSP       932 Response
   622 10.458325446  192.124.249.41    192.168.244.128   OCSP      2344 Response
```

```
▸ Frame 60: 427 bytes on wire (3416 bits), 427 bytes captured (3416 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 192.168.244.128, Dst: 117.18.237.29
▸ Transmission Control Protocol, Src Port: 57474, Dst Port: 80, Seq: 1, Ack: 1, Len: 371
▸ Hypertext Transfer Protocol
▸ Online Certificate Status Protocol
```

```
0000  00 04 00 01 00 06 00 0c  29 88 ce 79 00 00 08 00   ········ )··y····
0010  45 00 01 9b cf 0e 40 00  40 06 52 f5 c0 a8 f4 80   E·····@· @·R·····
0020  75 12 ed 1d e0 82 00 50  85 37 63 2f 16 b1 6d 15   u······P ·7c/··m·
0030  50 18 fa f0 18 e7 00 00  50 4f 53 54 20 2f 20 48   P······· POST / H
0040  54 54 50 2f 31 2e 31 0d  0a 48 6f 73 74 3a 20 6f   TTP/1.1· ·Host: o
```

Hypertext Transfer Protocol: Protocol   Packets: 789 · Displayed: 10 (1.3%)   Profile: Default


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http
```

```
No.   Time          Source           Destination      Protocol Length Info
   60 0.962334295   192.168.244.128  117.18.237.29    OCSP      427 Request
   62 0.966813590   117.18.237.29    192.168.244.128  OCSP      855 Response
  108 8.506096990   192.168.244.128  192.124.249.41   OCSP      419 Request
  110 8.860116551   192.124.249.41   192.168.244.128  OCSP     2344 Response
  514 9.890783108   192.168.244.128  104.18.31.182    OCSP      427 Request
  565 10.059797803  192.168.244.128  192.124.249.41   OCSP      419 Request
  591 10.102955036  192.168.244.128  192.124.249.41   OCSP      419 Request
  601 10.412728713  104.18.31.182    192.168.244.128  OCSP      996 Response
  613 10.416310720  192.124.249.41   192.168.244.128  OCSP      932 Response
  622 10.458325446  192.124.249.41   192.168.244.128  OCSP     2344 Response
```

```
▸ Frame 60: 427 bytes on wire (3416 bits), 427 bytes captured (3416 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 192.168.244.128, Dst: 117.18.237.29
▸ Transmission Control Protocol, Src Port: 57474, Dst Port: 80, Seq: 1, Ack: 1, Len: 371
▾ Hypertext Transfer Protocol
  ▸ POST / HTTP/1.1\r\n
    Host: ocsp.digicert.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/ocsp-request\r\n
```

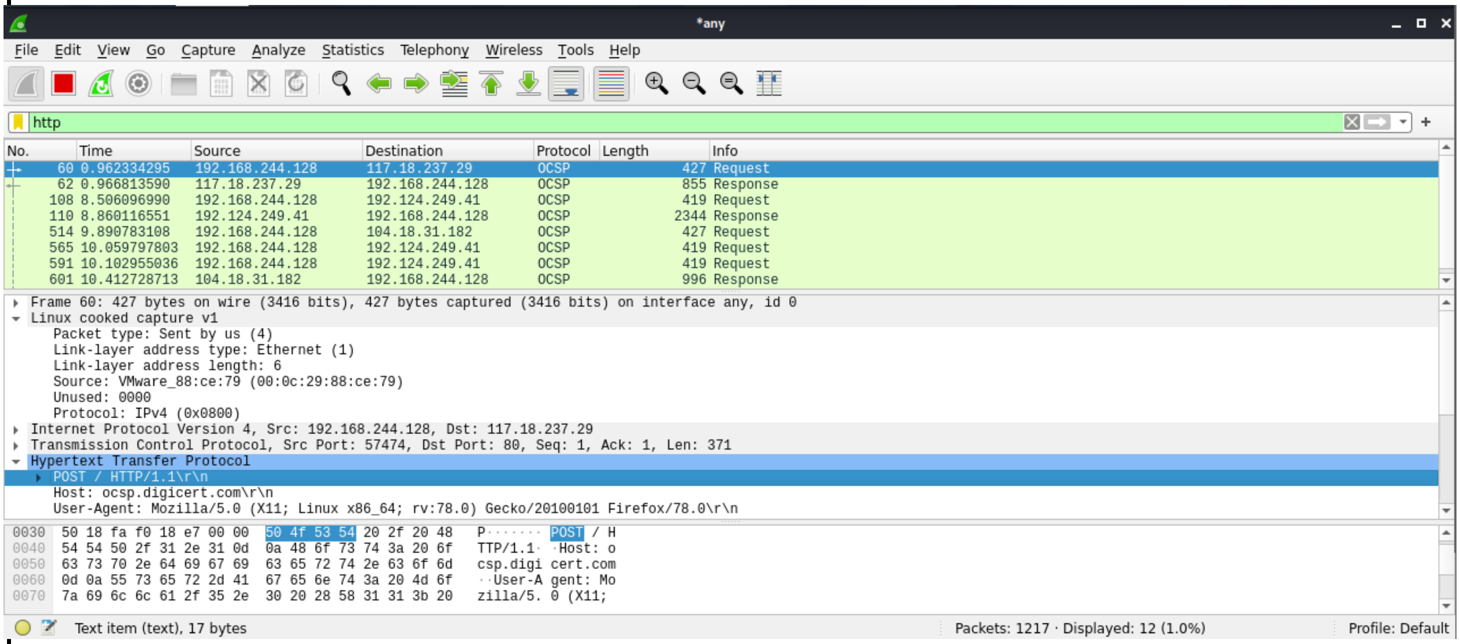
```
0000  00 04 00 01 00 06 00 0c  29 88 ce 79 00 00 08 00   ········ )··y····
0010  45 00 01 9b cf 0e 40 00  40 06 52 f5 c0 a8 f4 80   E·····@· @·R·····
0020  75 12 ed 1d e0 82 00 50  85 37 63 2f 16 b1 6d 15   u······P ·7c/··m·
0030  50 18 fa f0 18 e7 00 00  50 4f 53 54 20 2f 20 48   P······· POST / H
0040  54 54 50 2f 31 2e 31 0d  0a 48 6f 73 74 3a 20 6f   TTP/1.1· ·Host: o
```

Hypertext Transfer Protocol: Protocol | Packets: 1059 · Displayed: 12 (1.1%) | Profile: Default

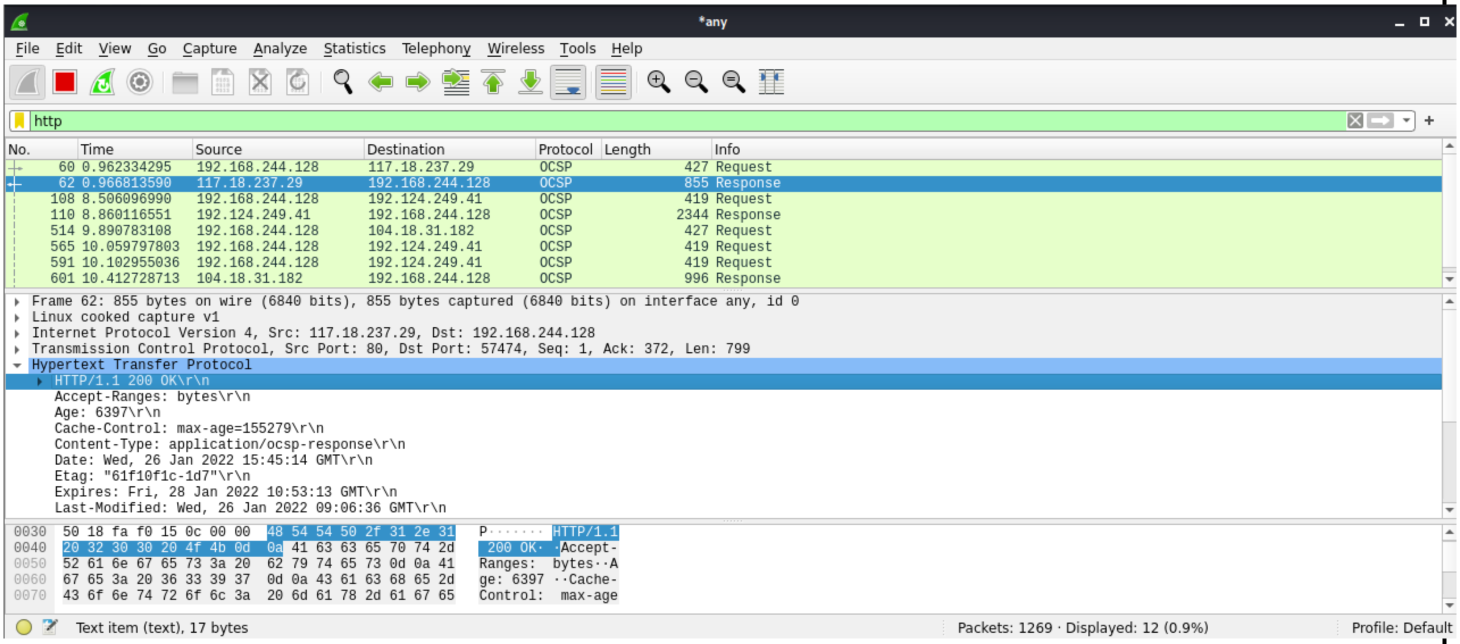
Step 3: Analyze the first (interaction of host to the web server) and second frame (response of server to the client). By analyzing the filtered frames, complete the table below:

Details	First Echo Request	First Echo Reply
Frame Number	60	3188
Source Port	57474	80
Destination Port	80	57474
Source IP address	192.168.244.128	117.18.237.29
Destination IP address	117.18.237.29	192.168.244.128
Source Ethernet Address	00:0c:29:88:ce:79	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	00:0c:29:88:ce:79
Internet Protocol Version	4	4
Time to Live (TTL) value	64	64

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



```
http

No.  Time          Source           Destination      Protocol Length Info
  60 0.962334295   192.168.244.128  117.18.237.29    OCSP        427 Request
  62 0.966813590   117.18.237.29    192.168.244.128  OCSP        855 Response
 108 8.506096990   192.168.244.128  192.124.249.41   OCSP        419 Request
 110 8.860116551   192.124.249.41   192.168.244.128  OCSP       2344 Response
 514 9.890783108   192.168.244.128  104.18.31.182    OCSP        427 Request
 565 10.059797803  192.168.244.128  192.124.249.41   OCSP        419 Request
 591 10.102955036  192.168.244.128  192.124.249.41   OCSP        419 Request
 601 10.412728713  104.18.31.182    192.168.244.128  OCSP        996 Response

Frame 60: 427 bytes on wire (3416 bits), 427 bytes captured (3416 bits) on interface any, id 0
Linux cooked capture v1
   Packet type: Sent by us (4)
   Link-layer address type: Ethernet (1)
   Link-layer address length: 6
   Source: VMware_88:ce:79 (00:0c:29:88:ce:79)
   Unused: 0000
   Protocol: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.244.128, Dst: 117.18.237.29
Transmission Control Protocol, Src Port: 57474, Dst Port: 80, Seq: 1, Ack: 1, Len: 371
Hypertext Transfer Protocol
   POST / HTTP/1.1\r\n
   Host: ocsp.digicert.com\r\n
   User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n

0030  50 18 fa f0 18 e7 00 00  50 4f 53 54 20 2f 20 48   P······· POST / H
0040  54 54 50 2f 31 2e 31 0d  0a 48 6f 73 74 3a 20 6f   TTP/1.1· ·Host: o
0050  63 73 70 2e 64 69 67 69  63 65 72 74 2e 63 6f 6d   csp.digi cert.com
0060  0d 0a 55 73 65 72 2d 41  67 65 6e 74 3a 20 4d 6f   ··User-A gent: Mo
0070  7a 69 6c 6c 61 2f 35 2e  30 20 28 58 31 31 3b 20   zilla/5. 0 (X11;

Text item (text), 17 bytes          Packets: 1217 · Displayed: 12 (1.0%)          Profile: Default
```


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

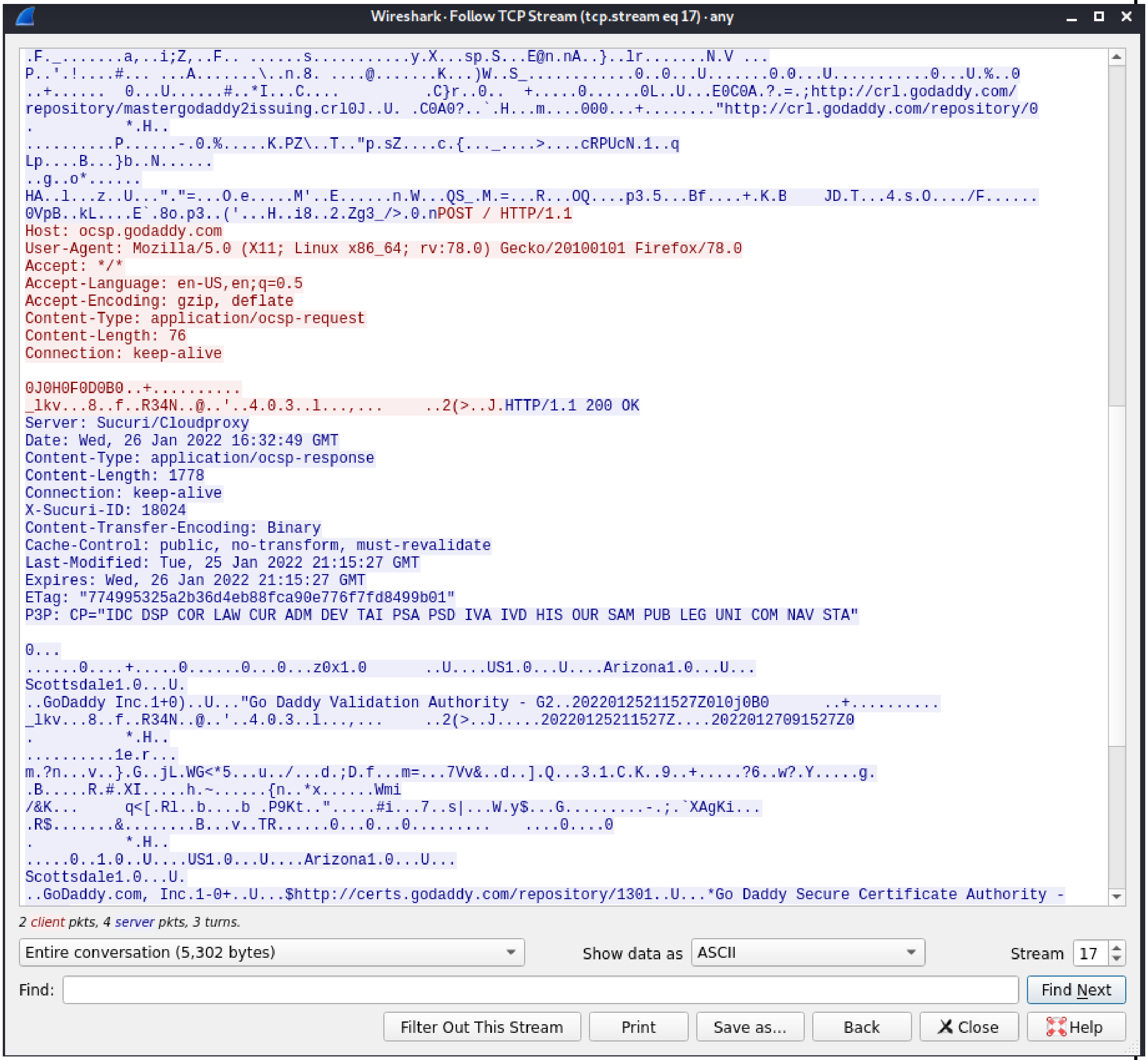
```
No.   Time          Source            Destination       Protocol Length  Info
   60 0.962334295   192.168.244.128   117.18.237.29     OCSP        427 Request
   62 0.966813590   117.18.237.29     192.168.244.128   OCSP        855 Response
  108 8.506096990   192.168.244.128   192.124.249.41    OCSP        419 Request
  110 8.860116551   192.124.249.41    192.168.244.128   OCSP       2344 Response
  514 9.890783108   192.168.244.128   104.18.31.182     OCSP        427 Request
  565 10.059797803  192.168.244.128   192.124.249.41    OCSP        419 Request
  591 10.102955036  192.168.244.128   192.124.249.41    OCSP        419 Request
  601 10.412728713  104.18.31.182     192.168.244.128   OCSP        996 Response

Frame 62: 855 bytes on wire (6840 bits), 855 bytes captured (6840 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 117.18.237.29, Dst: 192.168.244.128
Transmission Control Protocol, Src Port: 80, Dst Port: 57474, Seq: 1, Ack: 372, Len: 799
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Accept-Ranges: bytes\r\n
  Age: 6397\r\n
  Cache-Control: max-age=155279\r\n
  Content-Type: application/ocsp-response\r\n
  Date: Wed, 26 Jan 2022 15:45:14 GMT\r\n
  Etag: "61f10f1c-1d7"\r\n
  Expires: Fri, 28 Jan 2022 10:53:13 GMT\r\n
  Last-Modified: Wed, 26 Jan 2022 09:06:36 GMT\r\n

0030  50 18 fa f0 15 0c 00 00  48 54 54 50 2f 31 2e 31   P·······HTTP/1.1
0040  20 32 30 30 20 4f 4b 0d  0a 41 63 63 65 70 74 2d    200 OK··Accept-
0050  52 61 6e 67 65 73 3a 20  62 79 74 65 73 0d 0a 41   Ranges: bytes··A
0060  67 65 3a 20 36 33 39 37  0d 0a 43 61 63 68 65 2d   ge: 6397··Cache-
0070  43 6f 6e 74 72 6f 6c 3a  20 6d 61 78 2d 61 67 65   Control: max-age

Text item (text), 17 bytes          Packets: 1269 · Displayed: 12 (0.9%)          Profile: Default
```

HTTP Request		**HTTP Response**	
Get	HTTP/1.1 200 OK	Server	GWS
Host	detectportal.firefox.com	Content-Type	Text/plain
User-Agent	Mozilla/5.0 (X11; Linux x86_64)	Date	Wed, 26 Jan 2022 17:15:57 GMT
Accept-Language	en-US,en;q=0.5	Location	
Accept-Encoding	gzip, deflate	Content-Length	83
Connection	Keep-alive	Connection	Keep-alive

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.



```
.F._.......a,..i;Z,..F.. ......s...........y.X...sp.S...E@n.nA..}..lr.......N.V ...
P..'.!....#... ...A.......\..n.8. ....@.......K...)W..S_............0..0...U.......0.0...U...........0...U.%..0
..+...... 0...U......#..*I...C....          .C}r..0..  +.....0......0L..U...E0C0A.?.=.;http://crl.godaddy.com/
repository/mastergodaddy2issuing.crl0J..U. .C0A0?..`.H...m....000...+........"http://crl.godaddy.com/repository/0
.           *.H..
..........P......-.0.%.....K.PZ\..T.."p.sZ....c.{..._....>....cRPUcN.1..q
Lp....B...}b..N......
..g..o*......
HA..l...z..U...".".=...0.e.....M'..E......n.W...QS_.M.=...R...0Q....p3.5...Bf....+.K.B    JD.T...4.s.0..../F......
0VpB..kL....E`.8o.p3..('...H..i8..2.Zg3_/>.0.nPOST / HTTP/1.1
Host: ocsp.godaddy.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 76
Connection: keep-alive

0J0H0F0D0B0..+.........
_lkv...8..f..R34N..@..'..4.0.3..l...,...    ..2(>..J.HTTP/1.1 200 OK
Server: Sucuri/Cloudproxy
Date: Wed, 26 Jan 2022 16:32:49 GMT
Content-Type: application/ocsp-response
Content-Length: 1778
Connection: keep-alive
X-Sucuri-ID: 18024
Content-Transfer-Encoding: Binary
Cache-Control: public, no-transform, must-revalidate
Last-Modified: Tue, 25 Jan 2022 21:15:27 GMT
Expires: Wed, 26 Jan 2022 21:15:27 GMT
ETag: "774995325a2b36d4eb88fca90e776f7fd8499b01"
P3P: CP="IDC DSP COR LAW CUR ADM DEV TAI PSA PSD IVA IVD HIS OUR SAM PUB LEG UNI COM NAV STA"

0...
......0....+.....0......0...0...z0x1.0      ..U....US1.0...U....Arizona1.0...U...
Scottsdale1.0...U.
..GoDaddy Inc.1+0)..U..."Go Daddy Validation Authority - G2..20220125211527Z0l0j0B0      ..+..........
_lkv...8..f..R34N..@..'..4.0.3..l...,...    ..2(>..J.....20220125211527Z....20220127091527Z0
.           *.H..
..........1e.r...
m.?n...v..}.G..jL.WG<*5...u../...d.;D.f...m=...7Vv&..d..].Q...3.1.C.K..9..+.....?6..w?.Y.....g.
.B.....R.#.XI.....h.~......{n..*x......Wmi
/&K...      q<[.Rl..b....b .P9Kt..".....#i...7..s|...W.y$...G.........-.;.`XAgKi...
.R$.......&........B...v..TR......0...0...0.........    ....0....0
.           *.H..
.....0..1.0..U....US1.0...U....Arizona1.0...U...
Scottsdale1.0...U.
..GoDaddy.com, Inc.1-0+..U...$http://certs.godaddy.com/repository/1301..U...*Go Daddy Secure Certificate Authority -
```

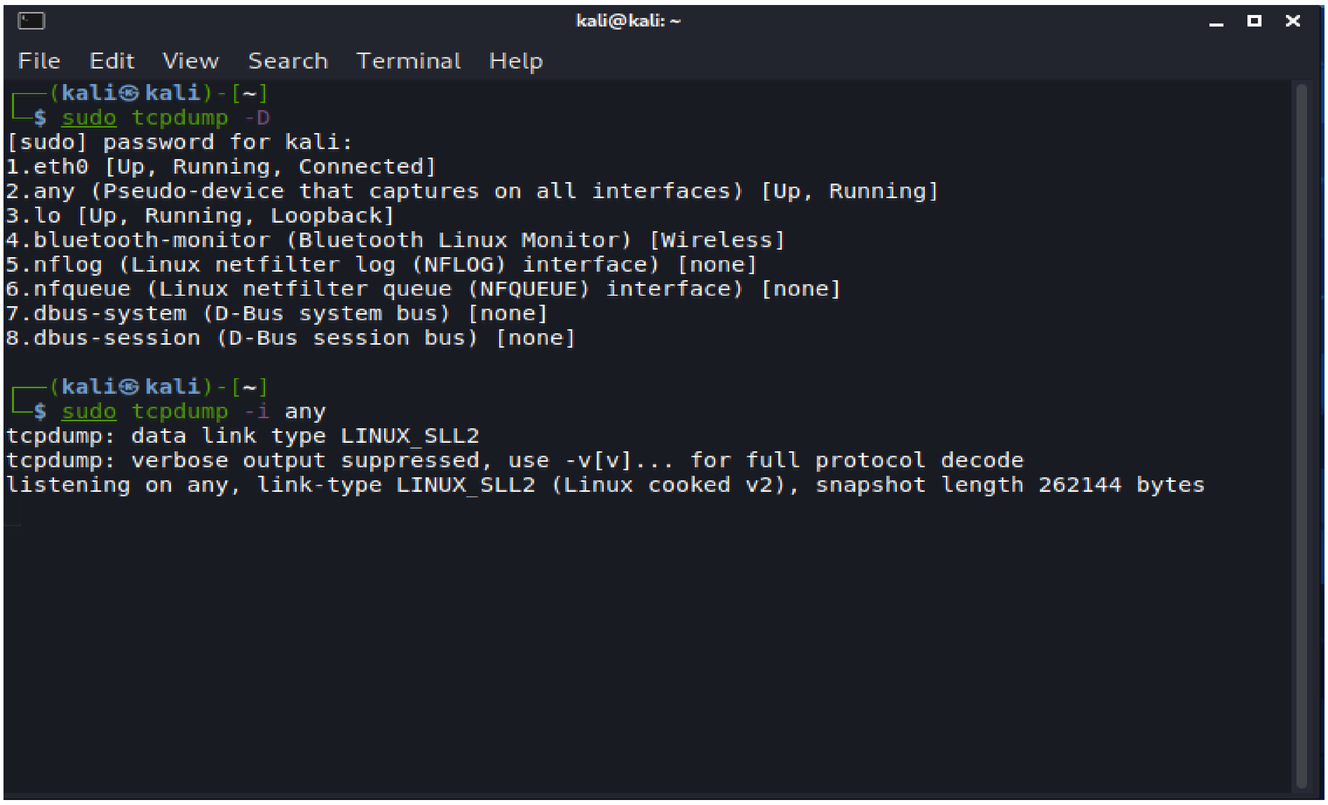
2 client pkts, 4 server pkts, 3 turns.

Entire conversation (5,302 bytes)   Show data as: ASCII   Stream: 17

Task 4: Capturing packets with tcpdump

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

sudo tcpdump -D

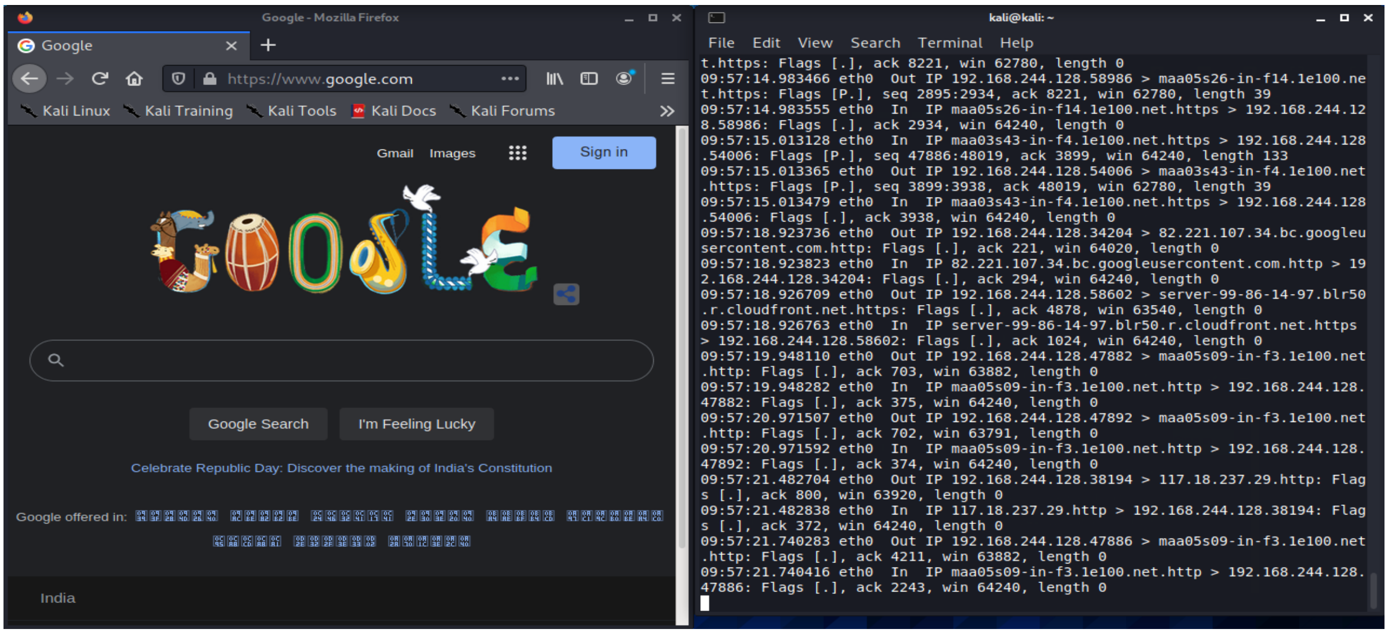


```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -D
[sudo] password for kali:
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [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]

┌──(kali㉿kali)-[~]
└─$ 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
```

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

sudo tcpdump -i any

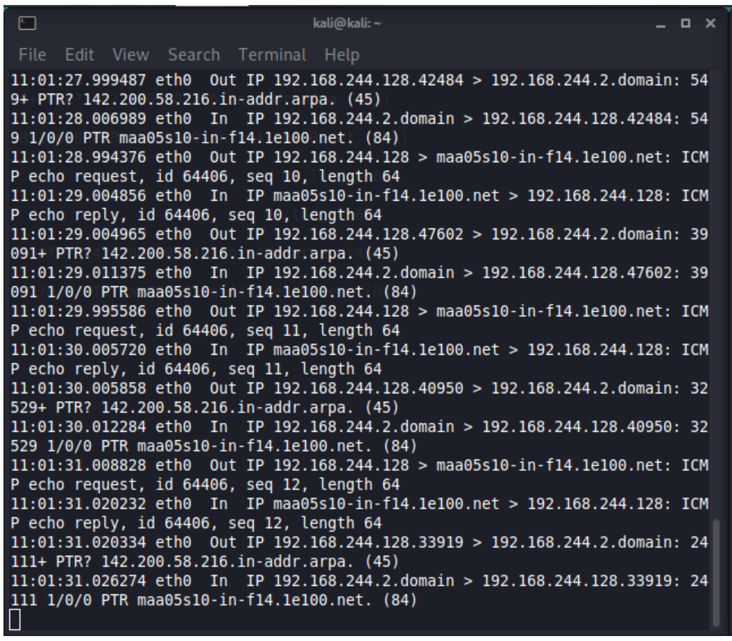
Note: Perform some pinging operation while giving above command. Also type www.google.com in browser.



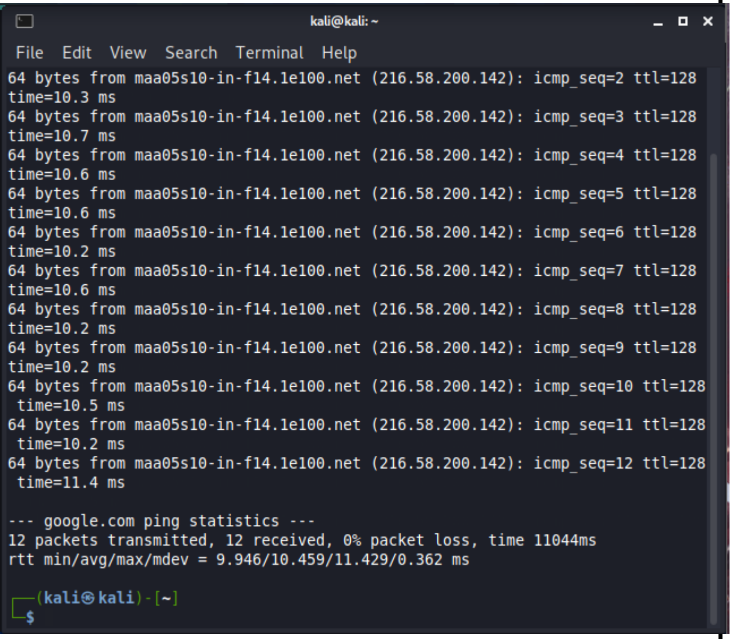
```
t.https: Flags [.], ack 8221, win 62780, length 0
09:57:14.983466 eth0  Out IP 192.168.244.128.58986 > maa05s26-in-f14.1e100.ne
t.https: Flags [P.], seq 2895:2934, ack 8221, win 62780, length 39
09:57:14.983555 eth0  In  IP maa05s26-in-f14.1e100.net.https > 192.168.244.12
8.58986: Flags [.], ack 2934, win 64240, length 0
09:57:15.013128 eth0  In  IP maa03s43-in-f4.1e100.net.https > 192.168.244.128
.54006: Flags [P.], seq 47886:48019, ack 3899, win 64240, length 133
09:57:15.013365 eth0  Out IP 192.168.244.128.54006 > maa03s43-in-f4.1e100.net
.https: Flags [P.], seq 3899:3938, ack 48019, win 62780, length 39
09:57:15.013479 eth0  In  IP maa03s43-in-f4.1e100.net.https > 192.168.244.128
.54006: Flags [.], ack 3938, win 64240, length 0
09:57:18.923736 eth0  Out IP 192.168.244.128.34204 > 82.221.107.34.bc.googleu
sercontent.com.http: Flags [.], ack 221, win 64020, length 0
09:57:18.923823 eth0  In  IP 82.221.107.34.bc.googleusercontent.com.http > 19
2.168.244.128.34204: Flags [.], ack 294, win 64240, length 0
09:57:18.926709 eth0  Out IP 192.168.244.128.58602 > server-99-86-14-97.blr50
.r.cloudfront.net.https: Flags [.], ack 4878, win 63540, length 0
09:57:18.926763 eth0  In  IP server-99-86-14-97.blr50.r.cloudfront.net.https
> 192.168.244.128.58602: Flags [.], ack 1024, win 64240, length 0
09:57:19.948110 eth0  Out IP 192.168.244.128.47882 > maa05s09-in-f3.1e100.net
.http: Flags [.], ack 703, win 63882, length 0
09:57:19.948282 eth0  In  IP maa05s09-in-f3.1e100.net.http > 192.168.244.128.
47882: Flags [.], ack 375, win 64240, length 0
09:57:20.971507 eth0  Out IP 192.168.244.128.47892 > maa05s09-in-f3.1e100.net
.http: Flags [.], ack 702, win 63791, length 0
09:57:20.971592 eth0  In  IP maa05s09-in-f3.1e100.net.http > 192.168.244.128.
47892: Flags [.], ack 374, win 64240, length 0
09:57:21.482704 eth0  Out IP 192.168.244.128.38194 > 117.18.237.29.http: Flag
s [.], ack 800, win 63920, length 0
09:57:21.482838 eth0  In  IP 117.18.237.29.http > 192.168.244.128.38194: Flag
s [.], ack 372, win 64240, length 0
09:57:21.740283 eth0  Out IP 192.168.244.128.47886 > maa05s09-in-f3.1e100.net
.http: Flags [.], ack 4211, win 63882, length 0
09:57:21.740416 eth0  In  IP maa05s09-in-f3.1e100.net.http > 192.168.244.128.
47886: Flags [.], ack 2243, win 64240, length 0
```

Observation:

Step 3: Understand the output format.



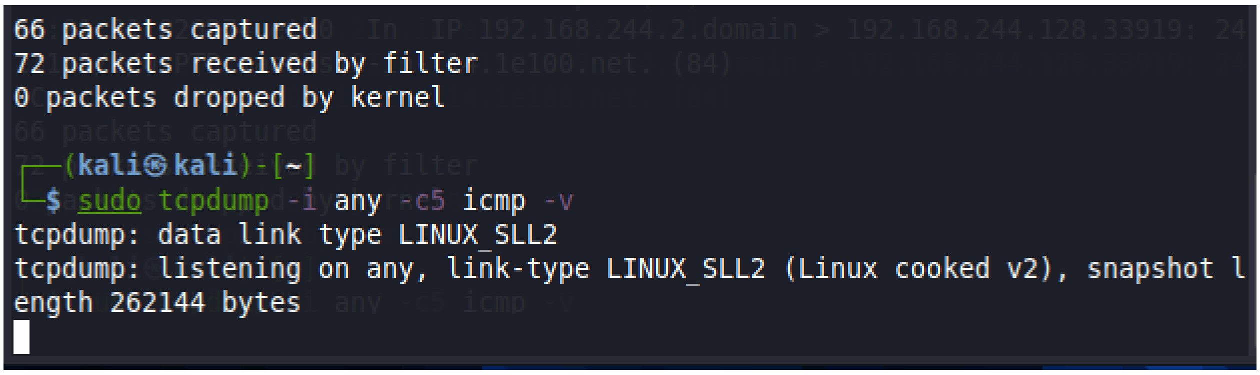
```
11:01:27.999487 eth0  Out IP 192.168.244.128.42484 > 192.168.244.2.domain: 54
9+ PTR? 142.200.58.216.in-addr.arpa. (45)
11:01:28.006989 eth0  In  IP 192.168.244.2.domain > 192.168.244.128.42484: 54
9 1/0/0 PTR maa05s10-in-f14.1e100.net. (84)
11:01:28.994376 eth0  Out IP 192.168.244.128 > maa05s10-in-f14.1e100.net: ICM
P echo request, id 64406, seq 10, length 64
11:01:29.004856 eth0  In  IP maa05s10-in-f14.1e100.net > 192.168.244.128: ICM
P echo reply, id 64406, seq 10, length 64
11:01:29.004965 eth0  Out IP 192.168.244.128.47602 > 192.168.244.2.domain: 39
091+ PTR? 142.200.58.216.in-addr.arpa. (45)
11:01:29.011375 eth0  In  IP 192.168.244.2.domain > 192.168.244.128.47602: 39
091 1/0/0 PTR maa05s10-in-f14.1e100.net. (84)
11:01:29.995586 eth0  Out IP 192.168.244.128 > maa05s10-in-f14.1e100.net: ICM
P echo request, id 64406, seq 11, length 64
11:01:30.005720 eth0  In  IP maa05s10-in-f14.1e100.net > 192.168.244.128: ICM
P echo reply, id 64406, seq 11, length 64
11:01:30.005858 eth0  Out IP 192.168.244.128.40950 > 192.168.244.2.domain: 32
529+ PTR? 142.200.58.216.in-addr.arpa. (45)
11:01:30.012284 eth0  In  IP 192.168.244.2.domain > 192.168.244.128.40950: 32
529 1/0/0 PTR maa05s10-in-f14.1e100.net. (84)
11:01:31.008828 eth0  Out IP 192.168.244.128 > maa05s10-in-f14.1e100.net: ICM
P echo request, id 64406, seq 12, length 64
11:01:31.020232 eth0  In  IP maa05s10-in-f14.1e100.net > 192.168.244.128: ICM
P echo reply, id 64406, seq 12, length 64
11:01:31.020334 eth0  Out IP 192.168.244.128.33919 > 192.168.244.2.domain: 24
111+ PTR? 142.200.58.216.in-addr.arpa. (45)
11:01:31.026274 eth0  In  IP 192.168.244.2.domain > 192.168.244.128.33919: 24
111 1/0/0 PTR maa05s10-in-f14.1e100.net. (84)
```



```
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=2 ttl=128
time=10.3 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=3 ttl=128
time=10.7 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=4 ttl=128
time=10.6 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=5 ttl=128
time=10.6 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=6 ttl=128
time=10.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=7 ttl=128
time=10.6 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=8 ttl=128
time=10.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=9 ttl=128
time=10.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=10 ttl=128
 time=10.5 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=11 ttl=128
 time=10.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=12 ttl=128
 time=11.4 ms

--- google.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11044ms
rtt min/avg/max/mdev = 9.946/10.459/11.429/0.362 ms

┌──(kali㉿kali)-[~]
└─$
```

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:

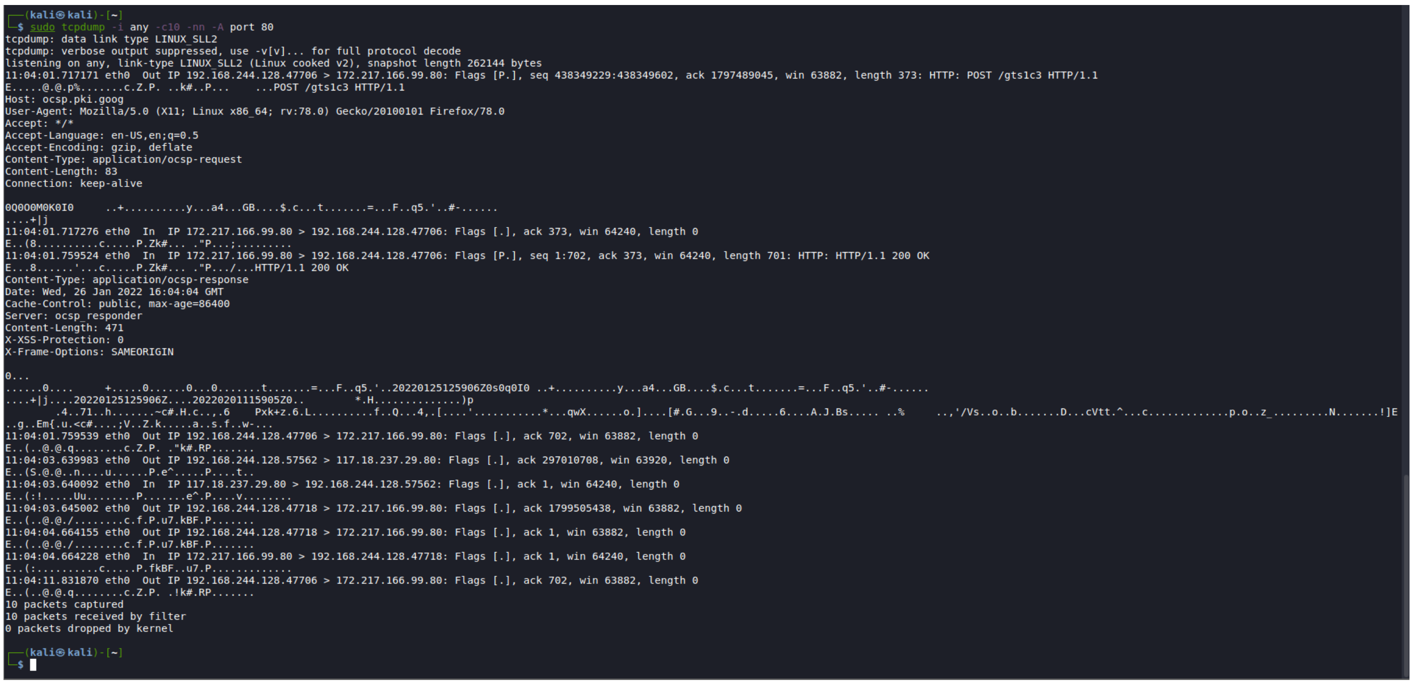
sudo tcpdump -i any -c5 icmp



```
66 packets captured
72 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i any -c5 icmp -v
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot l
ength 262144 bytes
```

Step 5: Check the packet content. For example, inspect the HTTP content of a web request like this:

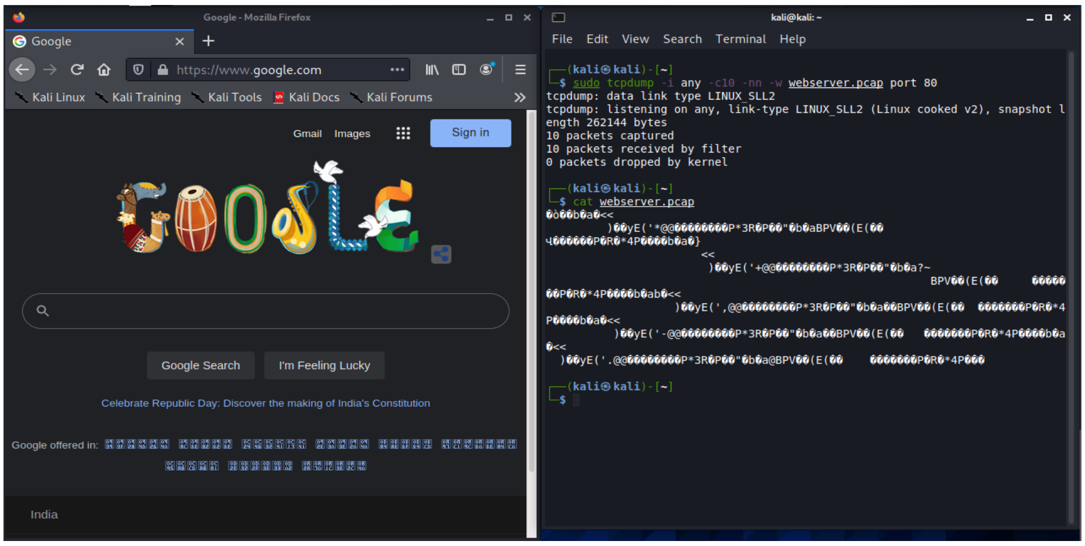
sudo tcpdump -i any -c10 -nn -A port 80



```
┌──(kali㉿kali)-[~]
└─$ 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
11:04:01.717171 eth0  Out IP 192.168.244.128.47706 > 172.217.166.99.80: Flags [P.], seq 438349229:438349602, ack 1797489045, win 63882, length 373: HTTP: POST /gts1c3 HTTP/1.1
E.....@.@.p%.......c.Z.P. ..k#..P...    ...POST /gts1c3 HTTP/1.1
Host: ocsp.pki.goog
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 83
Connection: keep-alive

0Q0O0M0K0I0     ..+.........y...a4...GB....$.c...t.......=...F..q5.'..#-......
....+|j
11:04:01.717276 eth0  In  IP 172.217.166.99.80 > 192.168.244.128.47706: Flags [.], ack 373, win 64240, length 0
E..(8.........c.....P.Zk#... ."P...;.........
11:04:01.759524 eth0  In  IP 172.217.166.99.80 > 192.168.244.128.47706: Flags [P.], seq 1:702, ack 373, win 64240, length 701: HTTP: HTTP/1.1 200 OK
E...8......'...c.....P.Zk#... ."P.../...HTTP/1.1 200 OK
Content-Type: application/ocsp-response
Date: Wed, 26 Jan 2022 16:04:04 GMT
Cache-Control: public, max-age=86400
Server: ocsp_responder
Content-Length: 471
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN

0...
......0....     +.....0......0...0.......t.......=...F..q5.'..20220125125906Z0s0q0I0 ..+.........y...a4...GB....$.c...t.......=...F..q5.'..#-......
....+|j....20220125125906Z....20220201115905Z0..        *.H..............)p
        .4..71..h.......~c#.H.c..,.6    Pxk+z.6.L.........f..Q...4,.[....'..........*...qwX......o.]....[#.G...9..-.d.....6....A.J.Bs..... ..%     ..,'/Vs..o..b.......D...cVtt.^...c............p.o..z_........N.......!]E
..g..Em{.u.<c#....;V..Z.k.....a..s.f..w-...
11:04:01.759539 eth0  Out IP 192.168.244.128.47706 > 172.217.166.99.80: Flags [.], ack 702, win 63882, length 0
E..(..@.@.q........c.Z.P. ."k#.RP.......
11:04:03.639983 eth0  Out IP 192.168.244.128.57562 > 117.18.237.29.80: Flags [.], ack 297010708, win 63920, length 0
E..(S.@.@..n....u......P.e^.....P....t..
11:04:03.640092 eth0  In  IP 117.18.237.29.80 > 192.168.244.128.57562: Flags [.], ack 1, win 64240, length 0
E..(:!.....Uu........P.......e^.P....v........
11:04:03.645002 eth0  Out IP 192.168.244.128.47718 > 172.217.166.99.80: Flags [.], ack 1799505438, win 63882, length 0
E..(..@.@./........c.f.P.u7.kBF.P.......
11:04:04.664155 eth0  Out IP 192.168.244.128.47718 > 172.217.166.99.80: Flags [.], ack 1, win 63882, length 0
E..(..@.@./........c.f.P.u7.kBF.P.......
11:04:04.664228 eth0  In  IP 172.217.166.99.80 > 192.168.244.128.47718: Flags [.], ack 1, win 64240, length 0
E..(:.........c.....P.fkBF..u7.P.............
11:04:11.831870 eth0  Out IP 192.168.244.128.47706 > 172.217.166.99.80: Flags [.], ack 702, win 63882, length 0
E..(..@.@.q........c.Z.P. .!k#.RP.......
10 packets captured
10 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ 
```

Step 6: To save packets to a file instead of displaying them on screen, use the option -w:

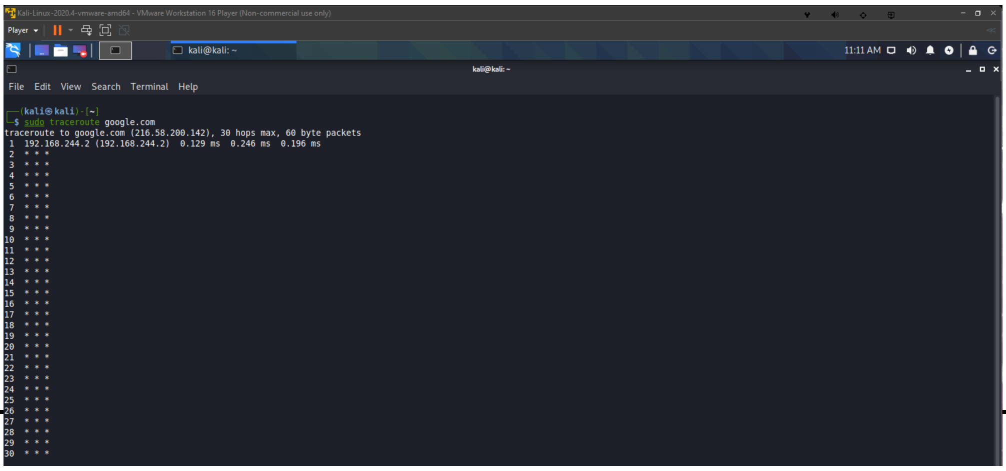
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80



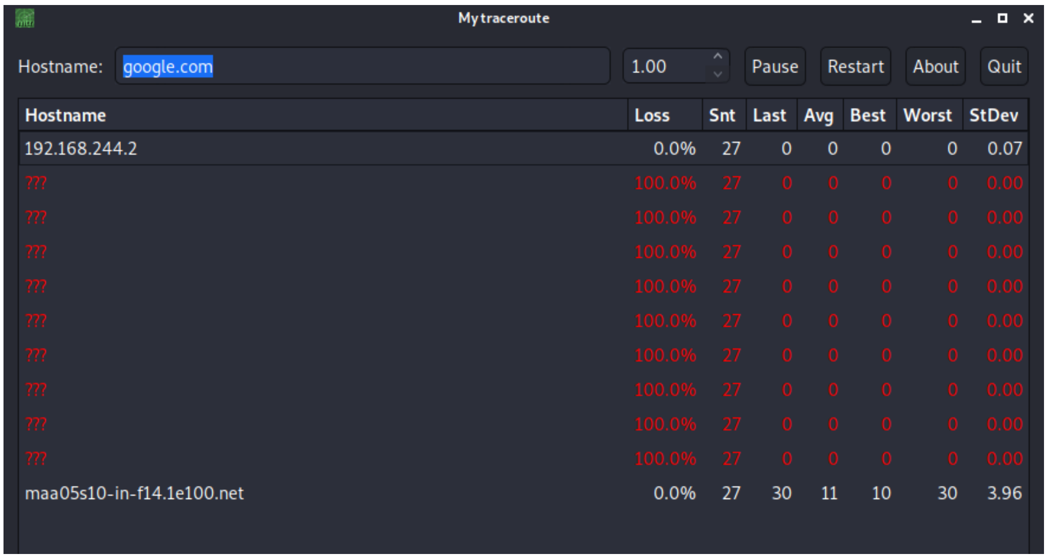
```
kali@kali: ~
File  Edit  View  Search  Terminal  Help

┌──(kali㉿kali)-[~]
└─$ 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 l
ength 262144 bytes
10 packets captured
10 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ cat webserver.pcap
�ò��b�a�<<
         )��yE('*@@���������P*3R�P��"�b�aBPV��(E(��
Ч������P�R�*4P����b�a�}
                             <<
                              )��yE('+@@���������P*3R�P��"�b�a?~
                                                              BPV��(E(��     �����
��P�R�*4P����b�ab�<<
                        )��yE(',@@���������P*3R�P��"�b�a��BPV��(E(��  �������P�R�*4
P����b�a�<<
          )��yE('-@@���������P*3R�P��"�b�a��BPV��(E(��   �������P�R�*4P����b�a
�<<
  )��yE('.@@���������P*3R�P��"�b�a@BPV��(E(��    �������P�R�*4P���

┌──(kali㉿kali)-[~]
└─$ 
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.


Kali-Linux-2020.4-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~
11:11 AM

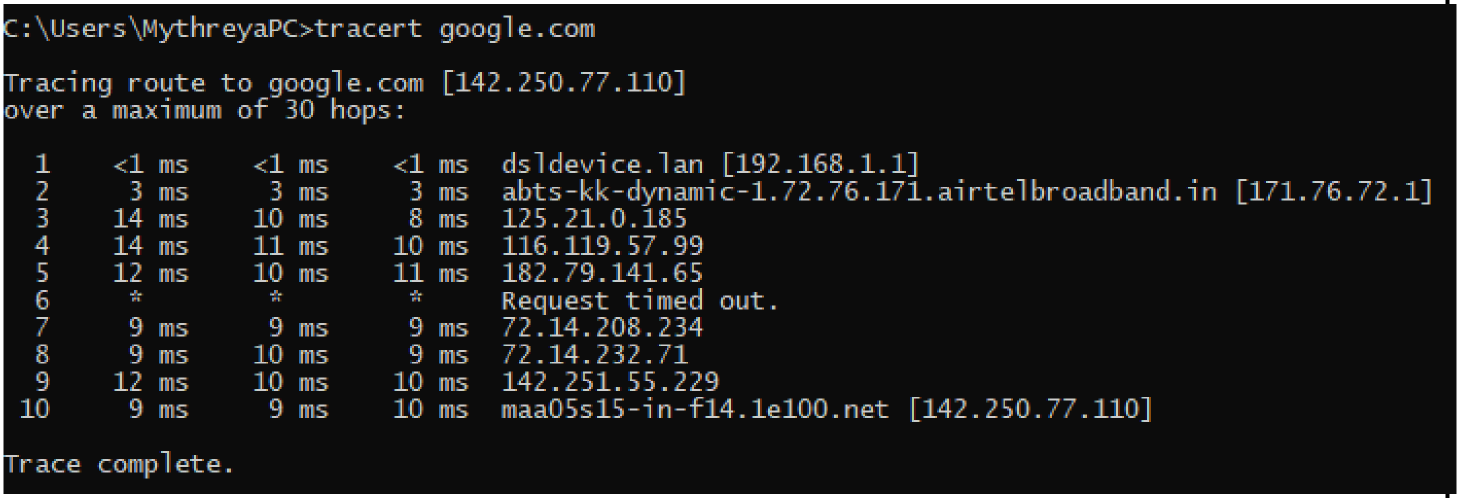
```
kali@kali: ~
File  Edit  View  Search  Terminal  Help

┌──(kali㉿kali)-[~]
└─$ sudo traceroute google.com
traceroute to google.com (216.58.200.142), 30 hops max, 60 byte packets
 1  192.168.244.2 (192.168.244.2)  0.129 ms  0.246 ms  0.196 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  * * *
```



Mytraceroute

Hostname: google.com 1.00 Pause Restart About Quit

Hostname	Loss	Snt	Last	Avg	Best	Worst	StDev
192.168.244.2	0.0%	27	0	0	0	0	0.07
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
???	100.0%	27	0	0	0	0	0.00
maa05s10-in-f14.1e100.net	0.0%	27	30	11	10	30	3.96



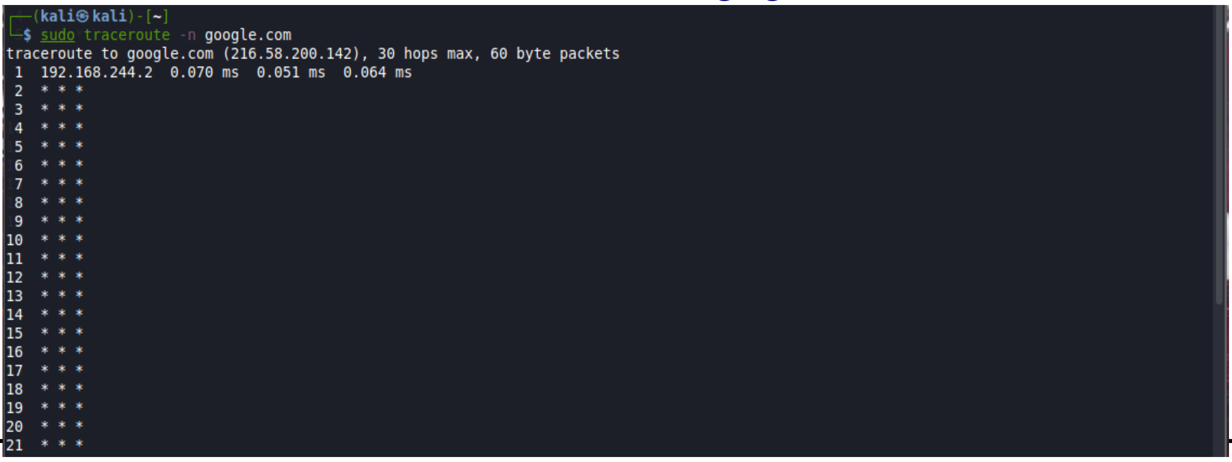
```
C:\Users\MythreyaPC>tracert google.com

Tracing route to google.com [142.250.77.110]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  dsldevice.lan [192.168.1.1]
  2     3 ms     3 ms     3 ms  abts-kk-dynamic-1.72.76.171.airtelbroadband.in [171.76.72.1]
  3    14 ms    10 ms     8 ms  125.21.0.185
  4    14 ms    11 ms    10 ms  116.119.57.99
  5    12 ms    10 ms    11 ms  182.79.141.65
  6     *        *        *     Request timed out.
  7     9 ms     9 ms     9 ms  72.14.208.234
  8     9 ms    10 ms     9 ms  72.14.232.71
  9    12 ms    10 ms    10 ms  142.251.55.229
 10     9 ms     9 ms    10 ms  maa05s15-in-f14.1e100.net [142.250.77.110]

Trace complete.
```

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

The destination address is (142.250.182.36), 30 hops max, 60 byte packets.

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the *-n* option

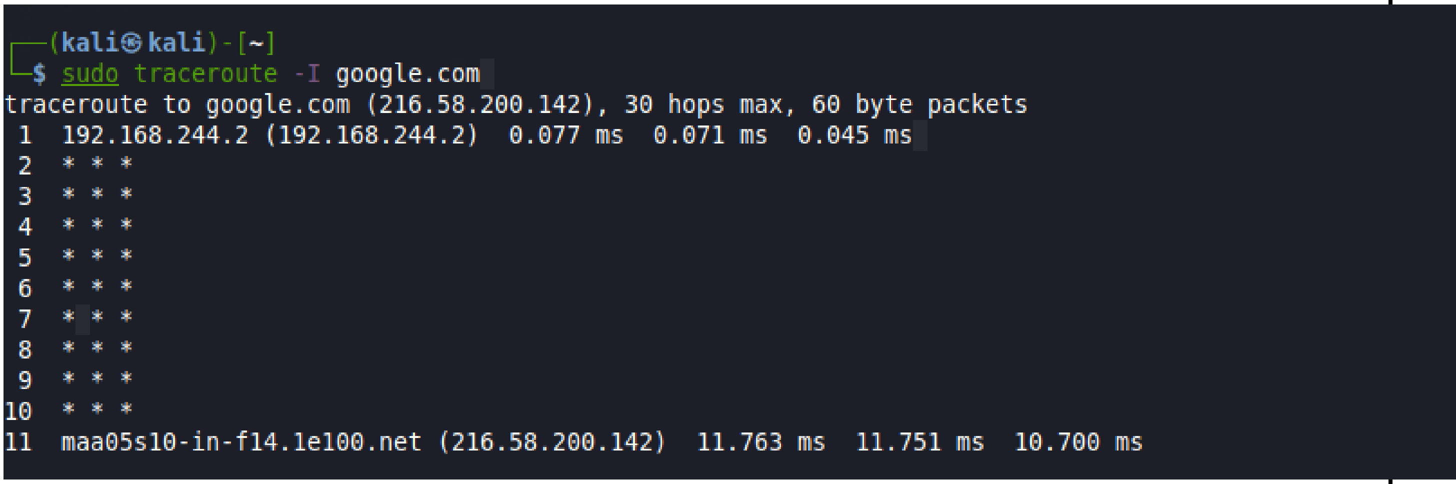
sudo traceroute -n www.google.com



```
┌──(kali㉿kali)-[~]
└─$ sudo traceroute -n google.com
traceroute to google.com (216.58.200.142), 30 hops max, 60 byte packets
 1  192.168.244.2  0.070 ms  0.051 ms  0.064 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
```

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

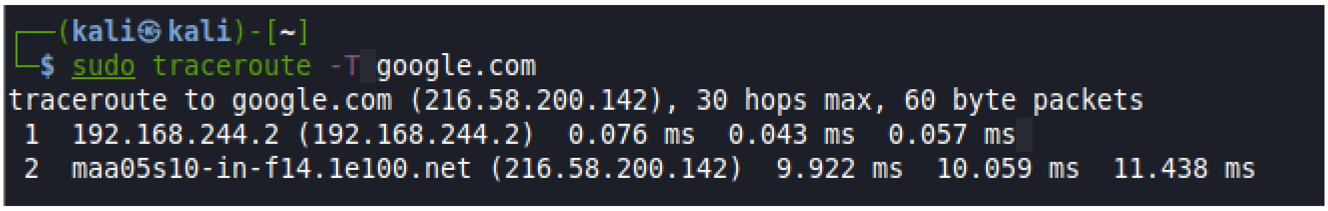
sudo traceroute -I www.google.com



```
┌──(kali㉿kali)-[~]
└─$ sudo traceroute -I google.com
traceroute to google.com (216.58.200.142), 30 hops max, 60 byte packets
 1  192.168.244.2 (192.168.244.2)  0.077 ms  0.071 ms  0.045 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  maa05s10-in-f14.1e100.net (216.58.200.142)  11.763 ms  11.751 ms  10.700 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

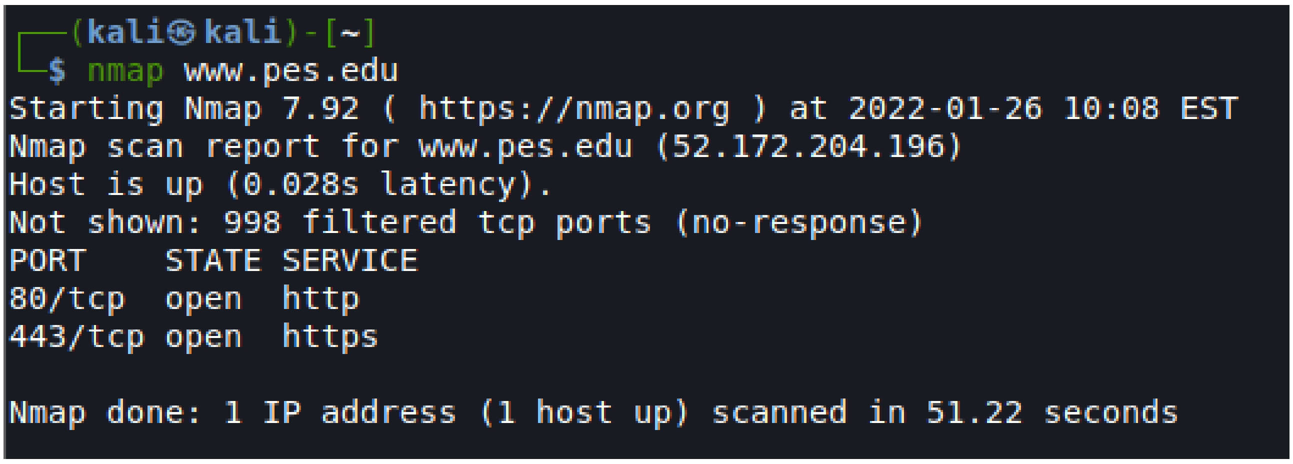


```
┌──(kali㉿kali)-[~]
└─$ sudo traceroute -T google.com
traceroute to google.com (216.58.200.142), 30 hops max, 60 byte packets
 1  192.168.244.2 (192.168.244.2)  0.076 ms  0.043 ms  0.057 ms
 2  maa05s10-in-f14.1e100.net (216.58.200.142)  9.922 ms  10.059 ms  11.438 ms
```

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



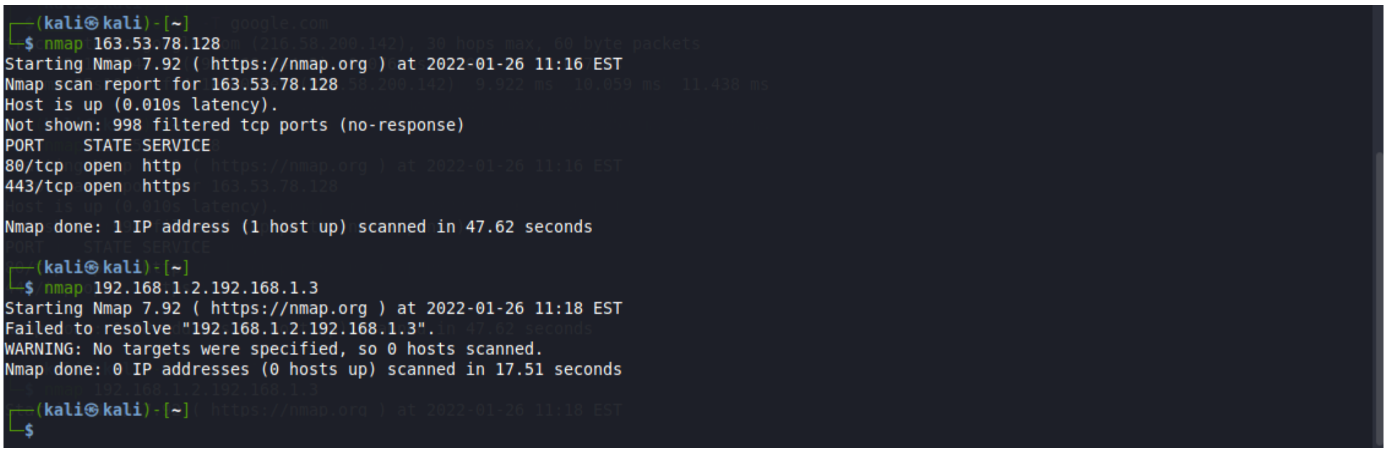
```
┌──(kali㉿kali)-[~]
└─$ nmap www.pes.edu
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-26 10:08 EST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.028s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 51.22 seconds
```

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

nmap 163.53.78.128

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

nmap 192.168.1.1 192.168.1.2 192.168.1.3



```
┌──(kali㉿kali)-[~]
└─$ nmap 163.53.78.128
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-26 11:16 EST
Nmap scan report for 163.53.78.128
Host is up (0.010s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 47.62 seconds

┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.2.192.168.1.3
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-26 11:18 EST
Failed to resolve "192.168.1.2.192.168.1.3".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 17.51 seconds

┌──(kali㉿kali)-[~]
└─$
```

Questions on above observations:

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

   Both uses Version 1.1

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

   Wed, 26 Jan 2022 16:32:12 GMT

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

   By using the flag "-c" followed by the number of packets to ping.
   Ex: ping -c 5 google.com
   pings google.com 5 times.

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

   By using "ip neigh"