Name: Shivam Sagpariya

Roll no. : 91800103191

Class – Tc2 / C

Practical: 3

1) Why do you use Wireshark? List benefits of Wireshark.

--Wireshark offers several benefits that make it appealing for everyday use. Aimed at both the up-and-coming and the expert packet analyst, it offers a variety of features to entice each. Let's examine Wireshark according to the criteria defined in Chapter 1 for selecting a packet-sniffing tool.

- Free software.
- Available for multiple platforms – Windows & UNIX.
- Can see detailed information about packets within a network.
- Not proprietary can be used on multiple vendors unlike Cisco Prime.

2) What is Packet Sniffer? How Packet Sniffers Work?

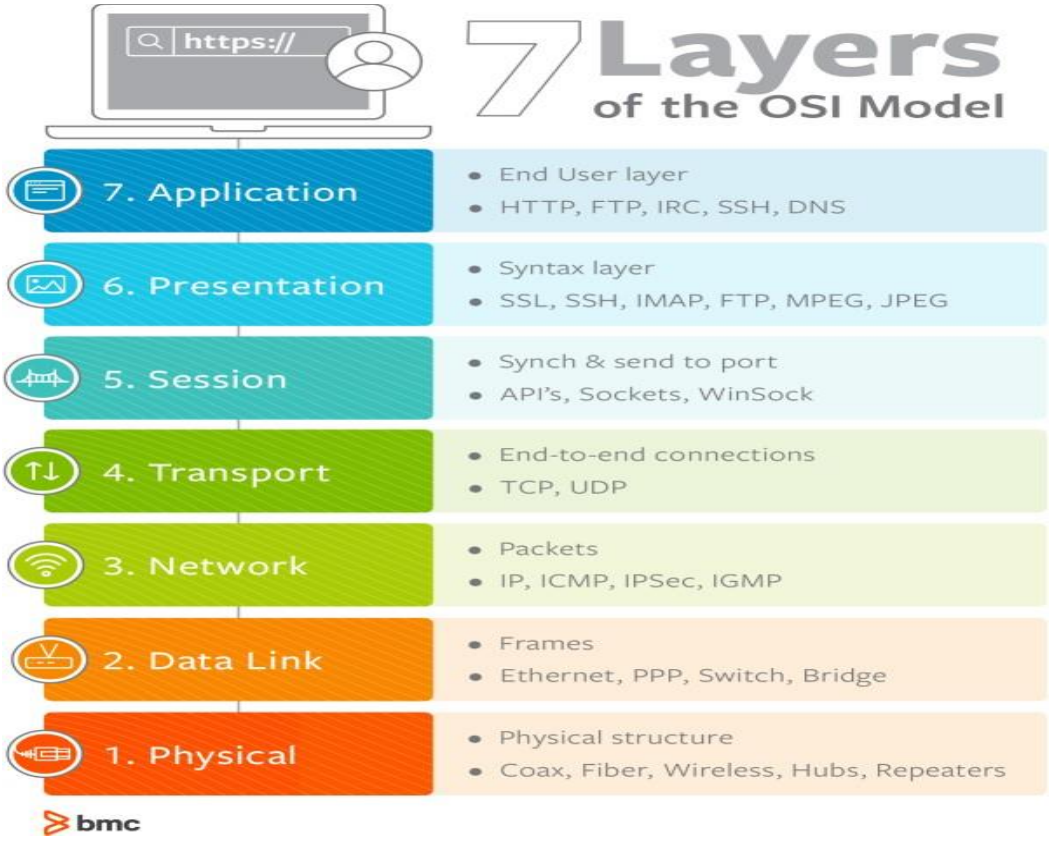
--Packet sniffing is the practice of gathering, collecting, and logging some or all packets that pass through a computer network, regardless of how the packet is addressed.

--A network is a collection of nodes, such as personal computers, servers, and networking hardware that are connected. The network connection allows data to be transferred between these devices. The connections can be physical with cables, or wireless with radio signals -- There are two main types of packet sniffers:

Hardware Packet Sniffers

Software Packet Sniffers

3) Draw the hierarchical view of the seven layers of the OSI model and write Typical Protocols Used at Each Layer of the OSI Model.



7 Layers of the OSI Model

7. Application
- End User layer
- HTTP, FTP, IRC, SSH, DNS

6. Presentation
- Syntax layer
- SSL, SSH, IMAP, FTP, MPEG, JPEG

5. Session
- Synch & send to port
- API's, Sockets, WinSock

4. Transport
- End-to-end connections
- TCP, UDP

3. Network
- Packets
- IP, ICMP, IPSec, IGMP

2. Data Link
- Frames
- Ethernet, PPP, Switch, Bridge

1. Physical
- Physical structure
- Coax, Fiber, Wireless, Hubs, Repeaters

4) What are the Wireshark Preferences?

Capture These preferences allow you to specify options related to the way packets are captured, including your default capture interface, whether to use promiscuous mode by default, and whether to update the Packet List pane in real time.

Appearance These preferences determine how Wireshark presents data. You can change most options here according to your personal preferences, including whether to save window positions, the layout of the three main panes, the placement of the scroll bar, the placement of the Packet List pane columns, the fonts used to display the captured data, and the background and foreground colors

Filter Expressions Later we will discuss how Wireshark allows you to filter traffic based on specific criteria. This section of the Preferences dialog allows you to create and manage those filters.

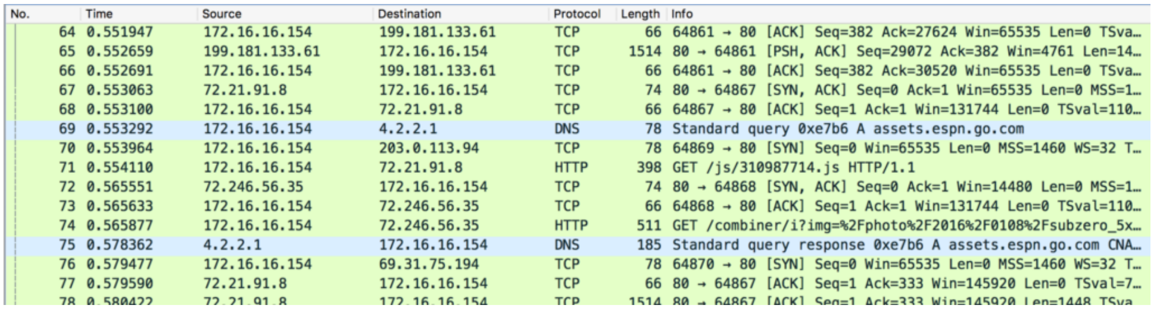
Name Resolution Through these preferences, you can activate features of Wireshark that allow it to resolve addresses into more recognizable names (including MAC, network, and transport name resolution) and specify the maximum number of concurrent name resolution requests.

Protocols This section allows you to manipulate options related to the capture and display of the various packets Wireshark is capable of decoding. Not every protocol has configurable preferences, but some have several options that can be changed. These options are best left at their defaults unless you have a specific reason to change them.

Statistics This section provides a few configurable options for Wireshark's statistical features. Advanced Settings that don't fit neatly into any of the previous categories can be found here. Editing the

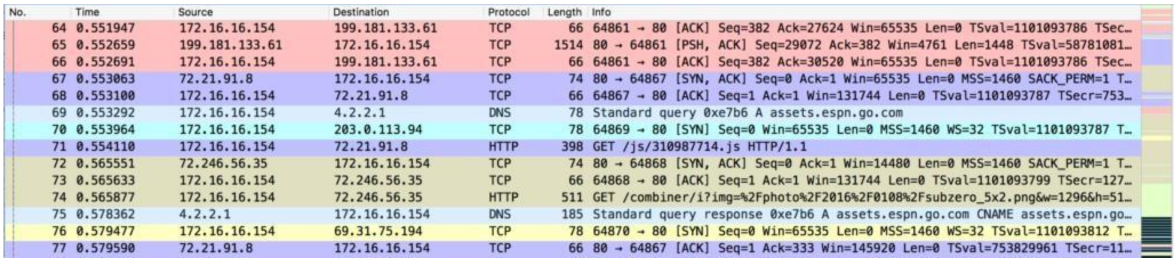
5)Describe the Packet Colour Coding in the Wireshark.

One of the biggest hindrances to analyzing packets occurs because so many things are happening simultaneously. Even something as simple as visiting a website will spawn connections to dozens of other hosts, sometimes with multiple conversations occurring per host. We want network communication to be fast, which means all of these connections are occurring at the same time. That's perfect for speed, but a nightmare for analysis. If you take a sample of twenty packets they might encompass a dozen or more individual conversations.



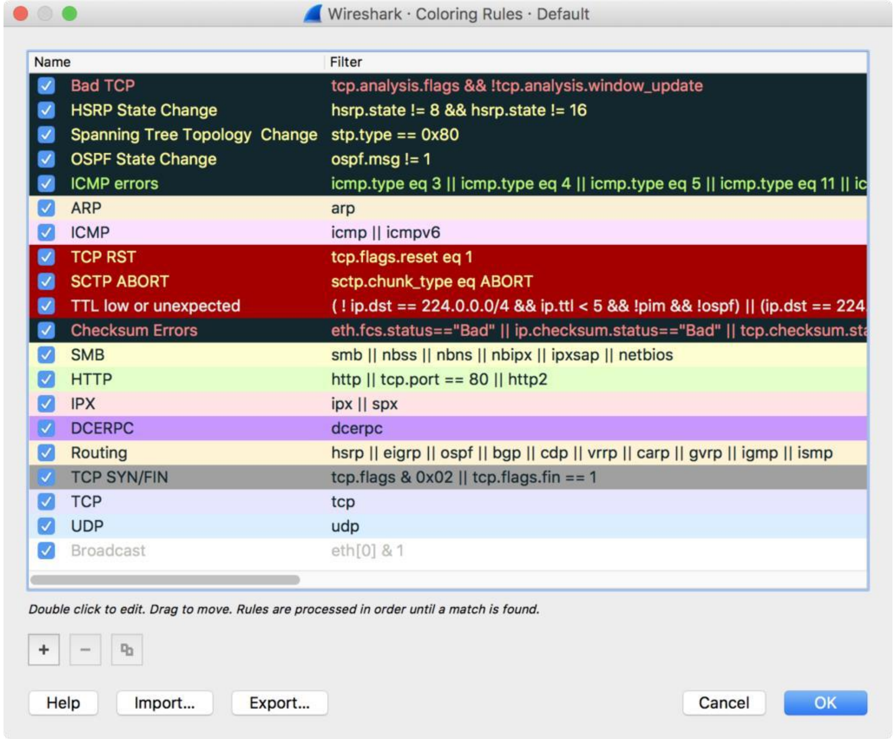
No.	Time	Source	Destination	Protocol	Length	Info
64	0.551947	172.16.16.154	199.181.133.61	TCP	66	64861 → 80 [ACK] Seq=382 Ack=27624 Win=65535 Len=0 TSva…
65	0.552659	199.181.133.61	172.16.16.154	TCP	1514	80 → 64861 [PSH, ACK] Seq=29072 Ack=382 Win=4761 Len=14…
66	0.552691	172.16.16.154	199.181.133.61	TCP	66	64861 → 80 [ACK] Seq=382 Ack=30520 Win=65535 Len=0 TSva…
67	0.553063	72.21.91.8	172.16.16.154	TCP	74	80 → 64867 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1…
68	0.553100	172.16.16.154	72.21.91.8	TCP	66	64867 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=110…
69	0.553292	172.16.16.154	4.2.2.1	DNS	78	Standard query 0xe7b6 A assets.espn.go.com
70	0.553964	172.16.16.154	203.0.113.94	TCP	78	64869 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 T…
71	0.554110	172.16.16.154	72.21.91.8	HTTP	398	GET /js/310987714.js HTTP/1.1
72	0.565551	72.246.56.35	172.16.16.154	TCP	74	80 → 64868 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1…
73	0.565633	172.16.16.154	72.246.56.35	TCP	66	64868 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=110…
74	0.565877	172.16.16.154	72.246.56.35	HTTP	511	GET /combiner/i?img=%2Fphoto%2F2016%2F0108%2Fsubzero_5x…
75	0.578362	4.2.2.1	172.16.16.154	DNS	185	Standard query response 0xe7b6 A assets.espn.go.com CNA…
76	0.579477	172.16.16.154	69.31.75.194	TCP	78	64870 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 T…
77	0.579590	72.21.91.8	172.16.16.154	TCP	66	80 → 64867 [ACK] Seq=1 Ack=333 Win=145920 Len=0 TSval=7…
78	0.580422	72.21.91.8	172.16.16.154	TCP	1514	80 → 64867 [ACK] Seq=1 Ack=333 Win=145920 Len=1448 TSva

Following individual streams would be an easy solution, but sometimes you want to see multiple conversations on the screen at once while being able to visually discern which conversation individual packets belong to. It's possible to determine that information from IP address and port numbers alone, but that's slow and error-prone. Wireshark provides great functionality to take advantage of how our mind processes visual input.



No.	Time	Source	Destination	Protocol	Length	Info
64	0.551947	172.16.16.154	199.181.133.61	TCP	66	64861 → 80 [ACK] Seq=382 Ack=27624 Win=65535 Len=0 TSval=1101093786 TSec…
65	0.552659	199.181.133.61	172.16.16.154	TCP	1514	80 → 64861 [PSH, ACK] Seq=29072 Ack=382 Win=4761 Len=1448 TSval=58781081…
66	0.552691	172.16.16.154	199.181.133.61	TCP	66	64861 → 80 [ACK] Seq=382 Ack=30520 Win=65535 Len=0 TSval=1101093786 TSec…
67	0.553063	72.21.91.8	172.16.16.154	TCP	74	80 → 64867 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1 T…
68	0.553100	172.16.16.154	72.21.91.8	TCP	66	64867 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=1101093787 TSecr=753…
69	0.553292	172.16.16.154	4.2.2.1	DNS	78	Standard query 0xe7b6 A assets.espn.go.com
70	0.553964	172.16.16.154	203.0.113.94	TCP	78	64869 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1101093787 T…
71	0.554110	172.16.16.154	72.21.91.8	HTTP	398	GET /js/310987714.js HTTP/1.1
72	0.565551	72.246.56.35	172.16.16.154	TCP	74	80 → 64868 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 T…
73	0.565633	172.16.16.154	72.246.56.35	TCP	66	64868 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=1101093799 TSecr=127…
74	0.565877	172.16.16.154	72.246.56.35	HTTP	511	GET /combiner/i?img=%2Fphoto%2F2016%2F0108%2Fsubzero_5x2.png&w=1296&h=51…
75	0.578362	4.2.2.1	172.16.16.154	DNS	185	Standard query response 0xe7b6 A assets.espn.go.com CNAME assets.espn.go…
76	0.579477	172.16.16.154	69.31.75.194	TCP	78	64870 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1101093812 T…
77	0.579590	72.21.91.8	172.16.16.154	TCP	66	80 → 64867 [ACK] Seq=1 Ack=333 Win=145920 Len=0 TSval=753829961 TSecr=11…

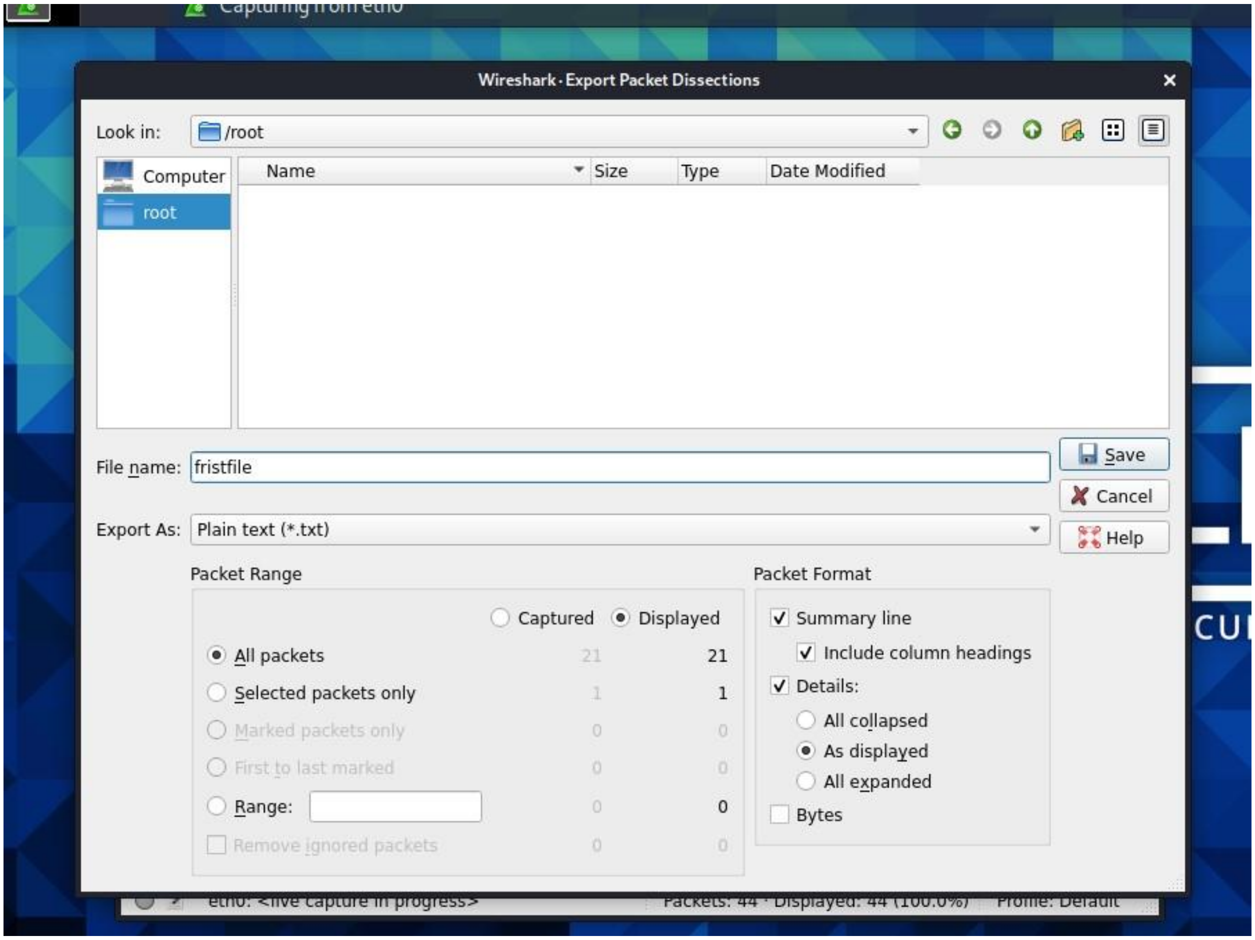
Wireshark color codes packets based on coloring rules. It comes with several of these built-in, but not everyone knows you can define your own custom coloring rules. To view the built-in coloring rules or to create your own, go to View > Coloring Rules.



Name	Filter
☑ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update
☑ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
☑ Spanning Tree Topology Change	stp.type == 0x80
☑ OSPF State Change	ospf.msg != 1
☑ ICMP errors	icmp.type eq 3 || icmp.type eq 4 || icmp.type eq 5 || icmp.type eq 11 || ic
☑ ARP	arp
☑ ICMP	icmp || icmpv6
☑ TCP RST	tcp.flags.reset eq 1
☑ SCTP ABORT	sctp.chunk_type eq ABORT
☑ TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf) || (ip.dst == 224
☑ Checksum Errors	eth.fcs.status=="Bad" || ip.checksum.status=="Bad" || tcp.checksum.st
☑ SMB	smb || nbss || nbns || nbipx || ipxsap || netbios
☑ HTTP	http || tcp.port == 80 || http2
☑ IPX	ipx || spx
☑ DCERPC	dcerpc
☑ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp || igmp || ismp
☑ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
☑ TCP	tcp
☑ UDP	udp
☑ Broadcast	eth[0] & 1

Double click to edit. Drag to move. Rules are processed in order until a match is found.

6) How to save and export the capture files? How to merge the capture files?



Capturing from eth0

Wireshark · Export Packet Dissections

Look in: /root

File name: fristfile

Export As: Plain text (*.txt)

Packet Range

	○ Captured	● Displayed
● All packets	21	21
○ Selected packets only	1	1
○ Marked packets only	0	0
○ First to last marked	0	0
○ Range:	0	0
□ Remove ignored packets	0	0

Packet Format

☑ Summary line

☑ Include column headings

☑ Details:

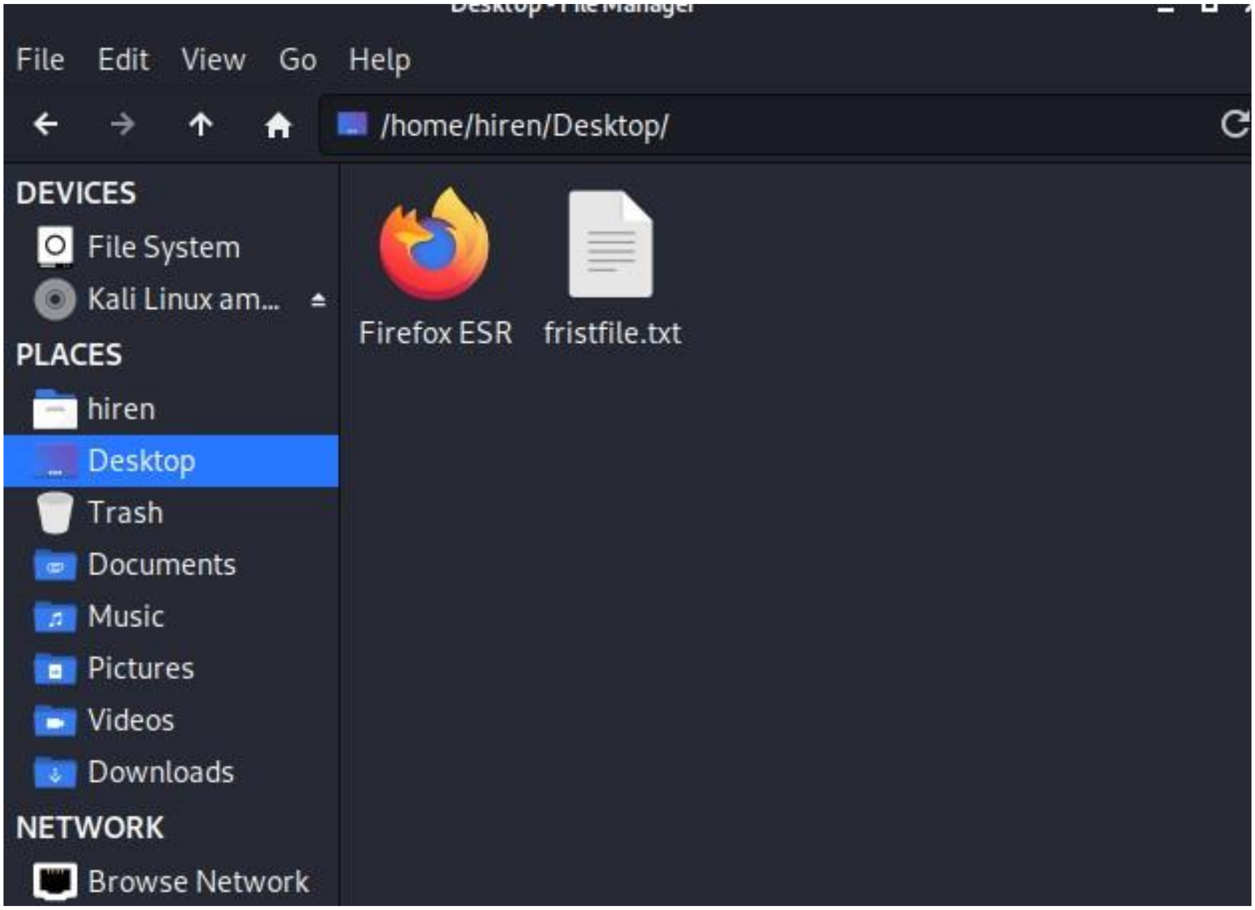
○ All collapsed

● As displayed

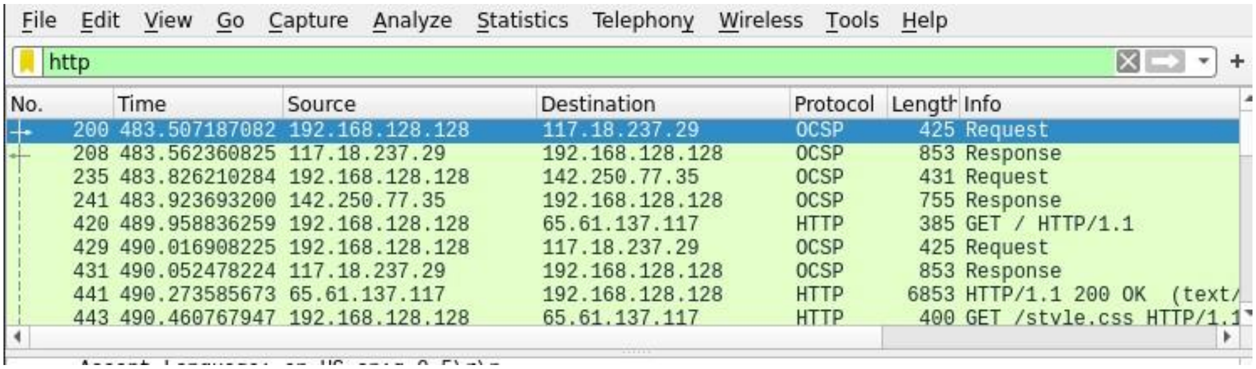
○ All expanded

□ Bytes

eth0: <live capture in progress> Packets: 44 · Displayed: 44 (100.0%) Profile: Default

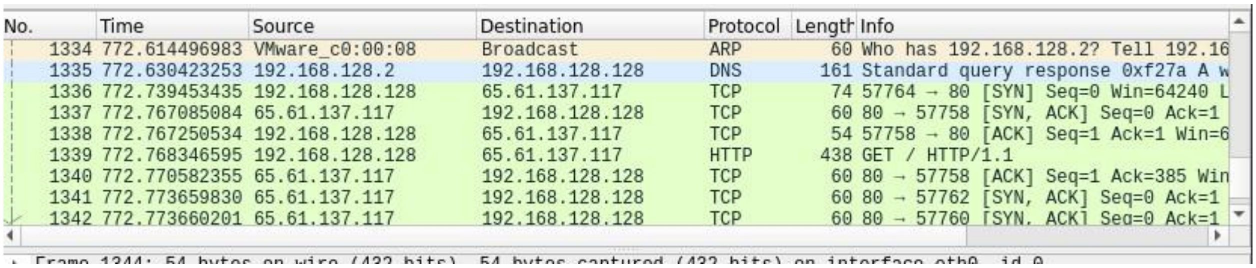


7) Capture the traffic from the website called www.altoromutual.com and observe the 3-Way Handshake process.Go to statistics and check flow graph and observe the 3-Way Handshake process. (Take Screenshot)


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

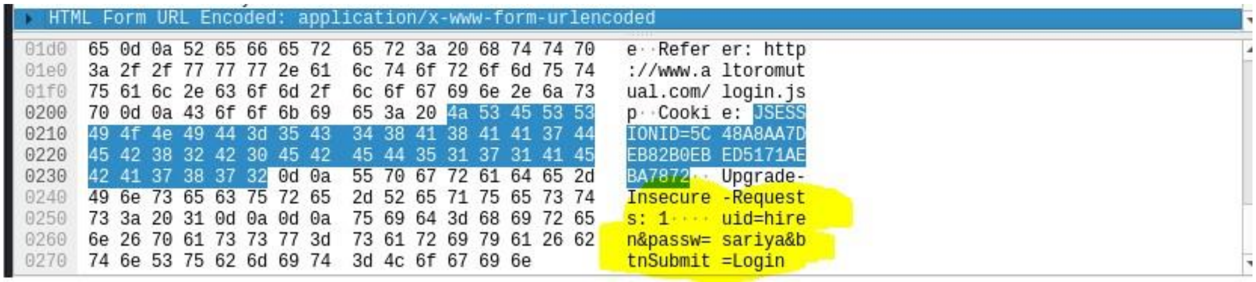
http

No.	Time	Source	Destination	Protocol	Length	Info
200	483.507187082	192.168.128.128	117.18.237.29	OCSP	425	Request
208	483.562360825	117.18.237.29	192.168.128.128	OCSP	853	Response
235	483.826210284	192.168.128.128	142.250.77.35	OCSP	431	Request
241	483.923693200	142.250.77.35	192.168.128.128	OCSP	755	Response
420	489.958836259	192.168.128.128	65.61.137.117	HTTP	385	GET / HTTP/1.1
429	490.016908225	192.168.128.128	117.18.237.29	OCSP	425	Request
431	490.052478224	117.18.237.29	192.168.128.128	OCSP	853	Response
441	490.273585673	65.61.137.117	192.168.128.128	HTTP	6853	HTTP/1.1 200 OK (text/
443	490.460767947	192.168.128.128	65.61.137.117	HTTP	400	GET /style.css HTTP/1.1



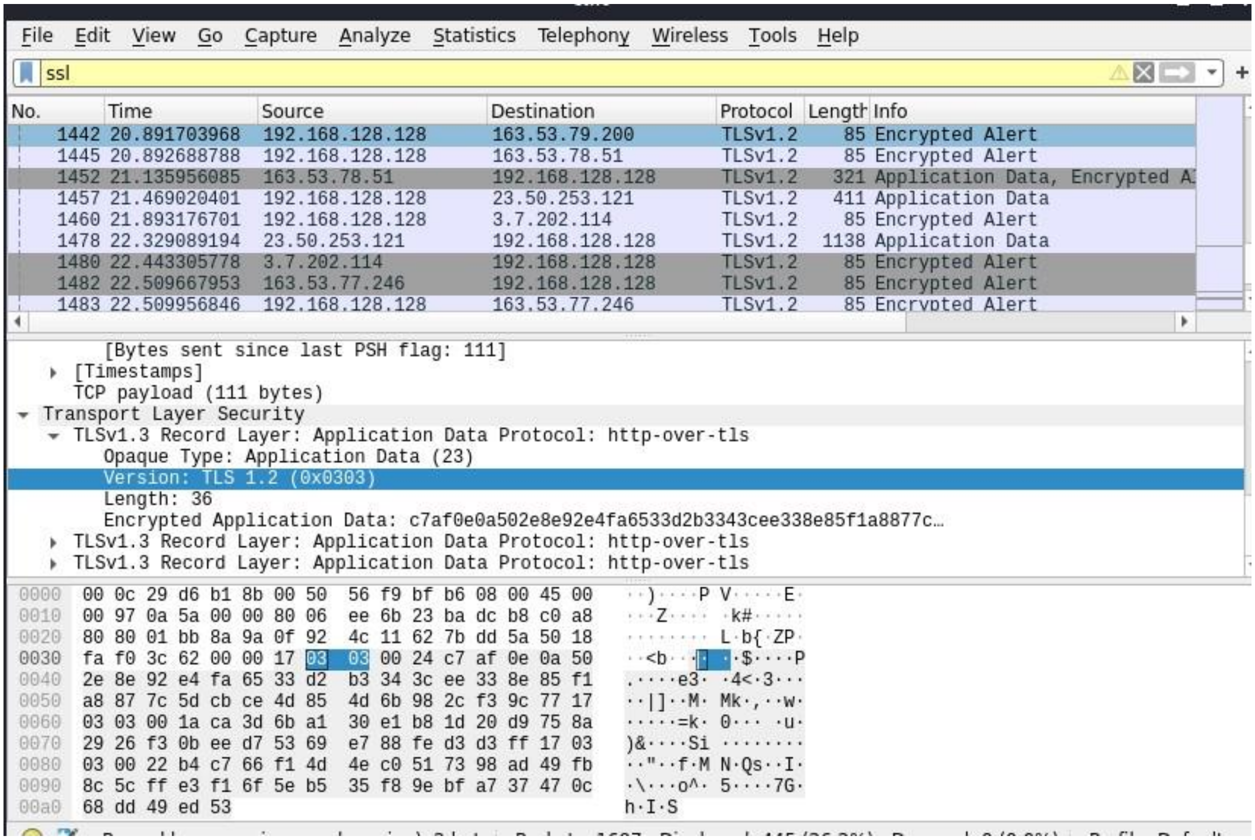
```
No.    Time         Source            Destination       Protocol Length Info
  1334 772.614496983 VMware_c0:00:08  Broadcast         ARP        60 Who has 192.168.128.2? Tell 192.16
  1335 772.630423253 192.168.128.2    192.168.128.128   DNS       161 Standard query response 0xf27a A w
  1336 772.739453435 192.168.128.128  65.61.137.117     TCP        74 57764 → 80 [SYN] Seq=0 Win=64240 L
  1337 772.767085084 65.61.137.117    192.168.128.128   TCP        60 80 → 57758 [SYN, ACK] Seq=0 Ack=1
  1338 772.767250534 192.168.128.128  65.61.137.117     TCP        54 57758 → 80 [ACK] Seq=1 Ack=1 Win=6
  1339 772.768346595 192.168.128.128  65.61.137.117     HTTP      438 GET / HTTP/1.1
  1340 772.770582355 65.61.137.117    192.168.128.128   TCP        60 80 → 57758 [ACK] Seq=1 Ack=385 Win
  1341 772.773659830 65.61.137.117    192.168.128.128   TCP        60 80 → 57762 [SYN, ACK] Seq=0 Ack=1
  1342 772.773660201 65.61.137.117    192.168.128.128   TCP        60 80 → 57760 [SYN, ACK] Seq=0 Ack=1
Frame 1344: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
```

8) Capture the traffic from the website called www.altoromutual.com do fake login in the field of user credentials.Observe the http traffic and check the fake password in the plain text format.



```
▸ HTML Form URL Encoded: application/x-www-form-urlencoded
01d0  65 0d 0a 52 65 66 65 72  65 72 3a 20 68 74 74 70   e··Refer er: http
01e0  3a 2f 2f 77 77 77 2e 61  6c 74 6f 72 6f 6d 75 74   ://www.a ltoromut
01f0  75 61 6c 2e 63 6f 6d 2f  6c 6f 67 69 6e 2e 6a 73   ual.com/ login.js
0200  70 0d 0a 43 6f 6f 6b 69  65 3a 20 4a 53 45 53 53   p··Cooki e: JSESS
0210  49 4f 4e 49 44 3d 35 43  34 38 41 38 41 41 37 44   IONID=5C 48A8AA7D
0220  45 42 38 32 42 30 45 42  45 44 35 31 37 31 41 45   EB82B0EB ED5171AE
0230  42 41 37 38 37 32 0d 0a  55 70 67 72 61 64 65 2d   BA7872·· Upgrade-
0240  49 6e 73 65 63 75 72 65  2d 52 65 71 75 65 73 74   Insecure -Request
0250  73 3a 20 31 0d 0a 0d 0a  75 69 64 3d 68 69 72 65   s: 1···· uid=hire
0260  6e 26 70 61 73 73 77 3d  73 61 72 69 79 61 26 62   n&passw= sariya&b
0270  74 6e 53 75 62 6d 69 74  3d 4c 6f 67 69 6e         tnSubmit =Login
```

9) Capture the traffic from any secure website and do fake login in the field of user credentials.Observe the SSL/TLS traffic and check the where password is stored and in which form?

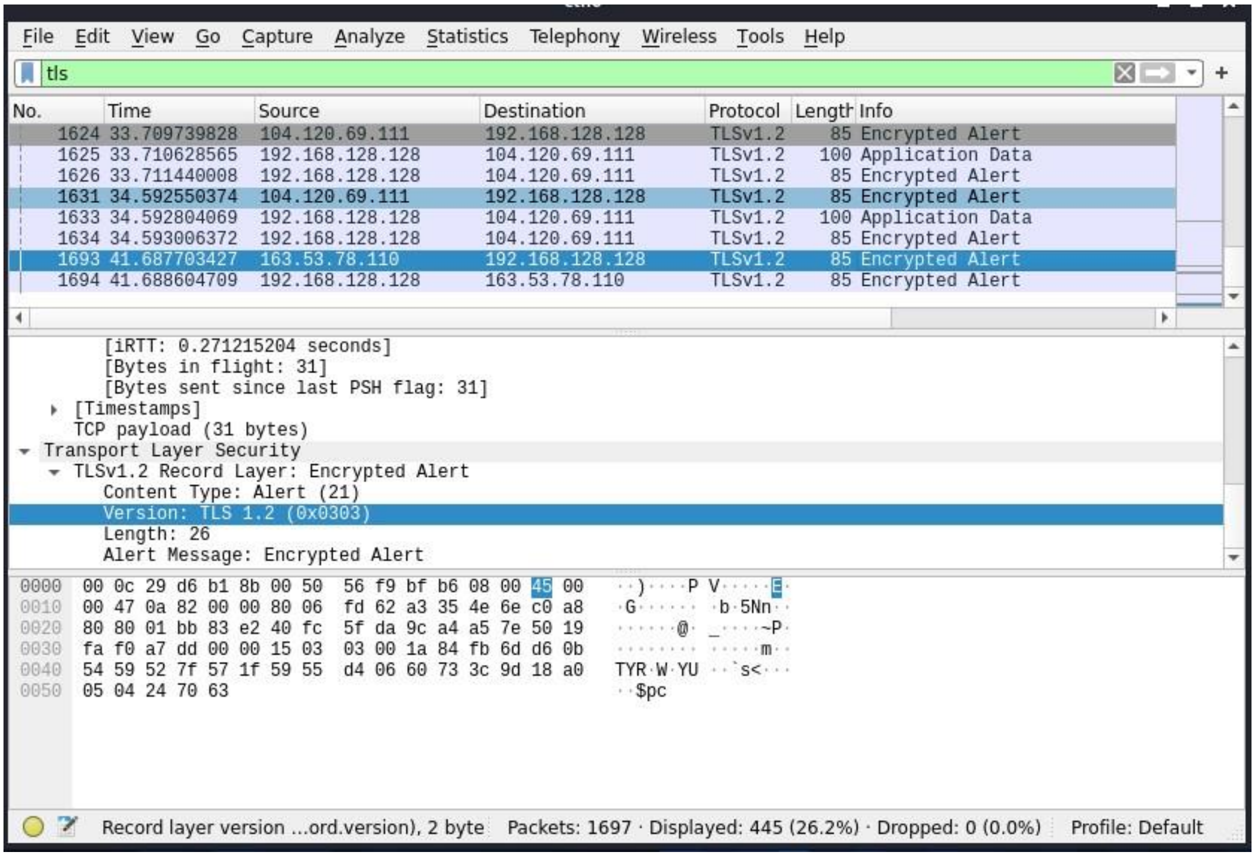


```
ssl
```

```
No.   Time         Source           Destination      Protocol Length Info
 1442 20.891703968 192.168.128.128  163.53.79.200    TLSv1.2    85 Encrypted Alert
 1445 20.892688788 192.168.128.128  163.53.78.51     TLSv1.2    85 Encrypted Alert
 1452 21.135956085 163.53.78.51     192.168.128.128  TLSv1.2   321 Application Data, Encrypted A
 1457 21.469020401 192.168.128.128  23.50.253.121    TLSv1.2   411 Application Data
 1460 21.893176701 192.168.128.128  3.7.202.114      TLSv1.2    85 Encrypted Alert
 1478 22.329089194 23.50.253.121    192.168.128.128  TLSv1.2  1138 Application Data
 1480 22.443305778 3.7.202.114      192.168.128.128  TLSv1.2    85 Encrypted Alert
 1482 22.509667953 163.53.77.246    192.168.128.128  TLSv1.2    85 Encrypted Alert
 1483 22.509956846 192.168.128.128  163.53.77.246    TLSv1.2    85 Encrypted Alert
```

```
      [Bytes sent since last PSH flag: 111]
  ▸ [Timestamps]
    TCP payload (111 bytes)
▾ Transport Layer Security
  ▾ TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 36
      Encrypted Application Data: c7af0e0a502e8e92e4fa6533d2b3343cee338e85f1a8877c…
  ▸ TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
  ▸ TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
```

```
0000  00 0c 29 d6 b1 8b 00 50  56 f9 bf b6 08 00 45 00   ··)····P V·····E·
0010  00 97 0a 5a 00 00 80 06  ee 6b 23 ba dc b8 c0 a8   ···Z····  ·k#·····
0020  80 80 01 bb 8a 9a 0f 92  4c 11 62 7b dd 5a 50 18   ········ L·b{·ZP·
0030  fa f0 3c 62 00 00 17 03  03 00 24 c7 af 0e 0a 50   ··<b····  ··$····P
0040  2e 8e 92 e4 fa 65 33 d2  b3 34 3c ee 33 8e 85 f1   .·····e3· ·4<·3···
0050  a8 87 7c 5d cb ce 4d 85  4d 6b 98 2c f3 9c 77 17   ··|]··M· Mk·,··w·
0060  03 03 00 1a ca 3d 6b a1  30 e1 b8 1d 20 d9 75 8a   ·····=k· 0··· ·u·
0070  29 26 f3 0b ee d7 53 69  e7 88 fe d3 d3 ff 17 03   )&····Si ········
0080  03 00 22 b4 c7 66 f1 4d  4e c0 51 73 98 ad 49 fb   ··"··f·M N·Qs··I·
0090  8c 5c ff e3 f1 6f 5e b5  35 f8 9e bf a7 37 47 0c   ·\···o^· 5····7G·
00a0  68 dd 49 ed 53                                      h·I·S
```



```
tls
```

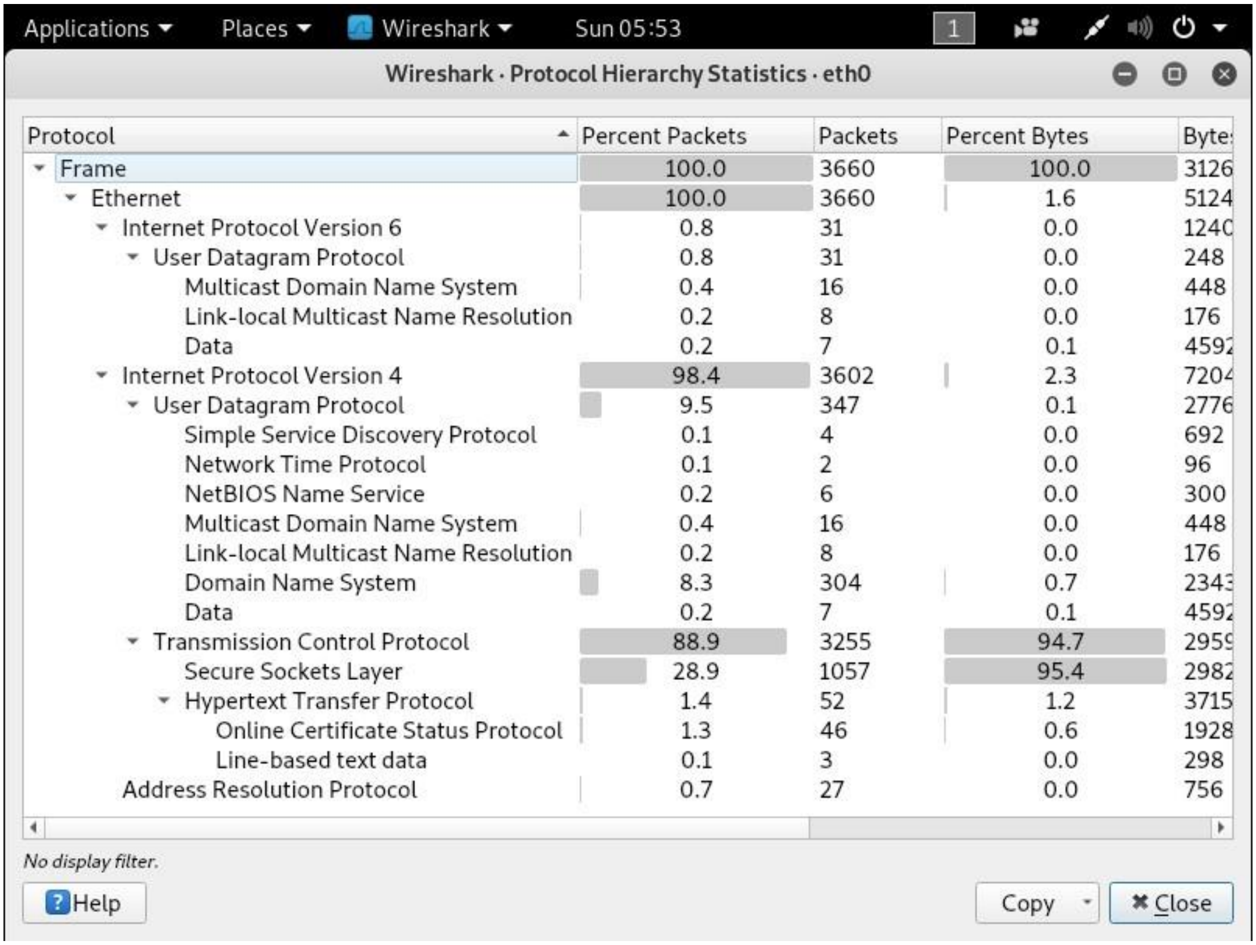
```
No.   Time         Source           Destination      Protocol Length Info
 1624 33.709739828 104.120.69.111   192.168.128.128  TLSv1.2    85 Encrypted Alert
 1625 33.710628565 192.168.128.128  104.120.69.111   TLSv1.2   100 Application Data
 1626 33.711440008 192.168.128.128  104.120.69.111   TLSv1.2    85 Encrypted Alert
 1631 34.592550374 104.120.69.111   192.168.128.128  TLSv1.2    85 Encrypted Alert
 1633 34.592804069 192.168.128.128  104.120.69.111   TLSv1.2   100 Application Data
 1634 34.593006372 192.168.128.128  104.120.69.111   TLSv1.2    85 Encrypted Alert
 1693 41.687703427 163.53.78.110    192.168.128.128  TLSv1.2    85 Encrypted Alert
 1694 41.688604709 192.168.128.128  163.53.78.110    TLSv1.2    85 Encrypted Alert
```

```
      [iRTT: 0.271215204 seconds]
      [Bytes in flight: 31]
      [Bytes sent since last PSH flag: 31]
  ▸ [Timestamps]
    TCP payload (31 bytes)
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Encrypted Alert
      Content Type: Alert (21)
      Version: TLS 1.2 (0x0303)
      Length: 26
      Alert Message: Encrypted Alert
```

```
0000  00 0c 29 d6 b1 8b 00 50  56 f9 bf b6 08 00 45 00   ··)····P V·····E·
0010  00 47 0a 82 00 00 80 06  fd 62 a3 35 4e 6e c0 a8   ·G······ ·b·5Nn··
0020  80 80 01 bb 83 e2 40 fc  5f da 9c a4 a5 7e 50 19   ······@· _····~P·
0030  fa f0 a7 dd 00 00 15 03  03 00 1a 84 fb 6d d6 0b   ········ ·····m··
0040  54 59 52 7f 57 1f 59 55  d4 06 60 73 3c 9d 18 a0   TYR·W·YU ··`s<···
0050  05 04 24 70 63                                      ··$pc
```

Record layer version ...ord.version), 2 byte Packets: 1697 · Displayed: 445 (26.2%) · Dropped: 0 (0.0%) Profile: Default

11) Capture the traffic from any secure or non-secure website and observe the Protocol Hierarchy Statistics.



Protocol	Percent Packets	Packets	Percent Bytes	Byte
Frame	100.0	3660	100.0	3126
Ethernet	100.0	3660	1.6	5124
Internet Protocol Version 6	0.8	31	0.0	1240
User Datagram Protocol	0.8	31	0.0	248
Multicast Domain Name System	0.4	16	0.0	448
Link-local Multicast Name Resolution	0.2	8	0.0	176
Data	0.2	7	0.1	4592
Internet Protocol Version 4	98.4	3602	2.3	7204
User Datagram Protocol	9.5	347	0.1	2776
Simple Service Discovery Protocol	0.1	4	0.0	692
Network Time Protocol	0.1	2	0.0	96
NetBIOS Name Service	0.2	6	0.0	300
Multicast Domain Name System	0.4	16	0.0	448
Link-local Multicast Name Resolution	0.2	8	0.0	176
Domain Name System	8.3	304	0.7	2343
Data	0.2	7	0.1	4592
Transmission Control Protocol	88.9	3255	94.7	2959
Secure Sockets Layer	28.9	1057	95.4	2982
Hypertext Transfer Protocol	1.4	52	1.2	3715
Online Certificate Status Protocol	1.3	46	0.6	1928
Line-based text data	0.1	3	0.0	298
Address Resolution Protocol	0.7	27	0.0	756

12) For which protocols stream can be observed? What is the use of that stream? Demonstrate with the proper example.

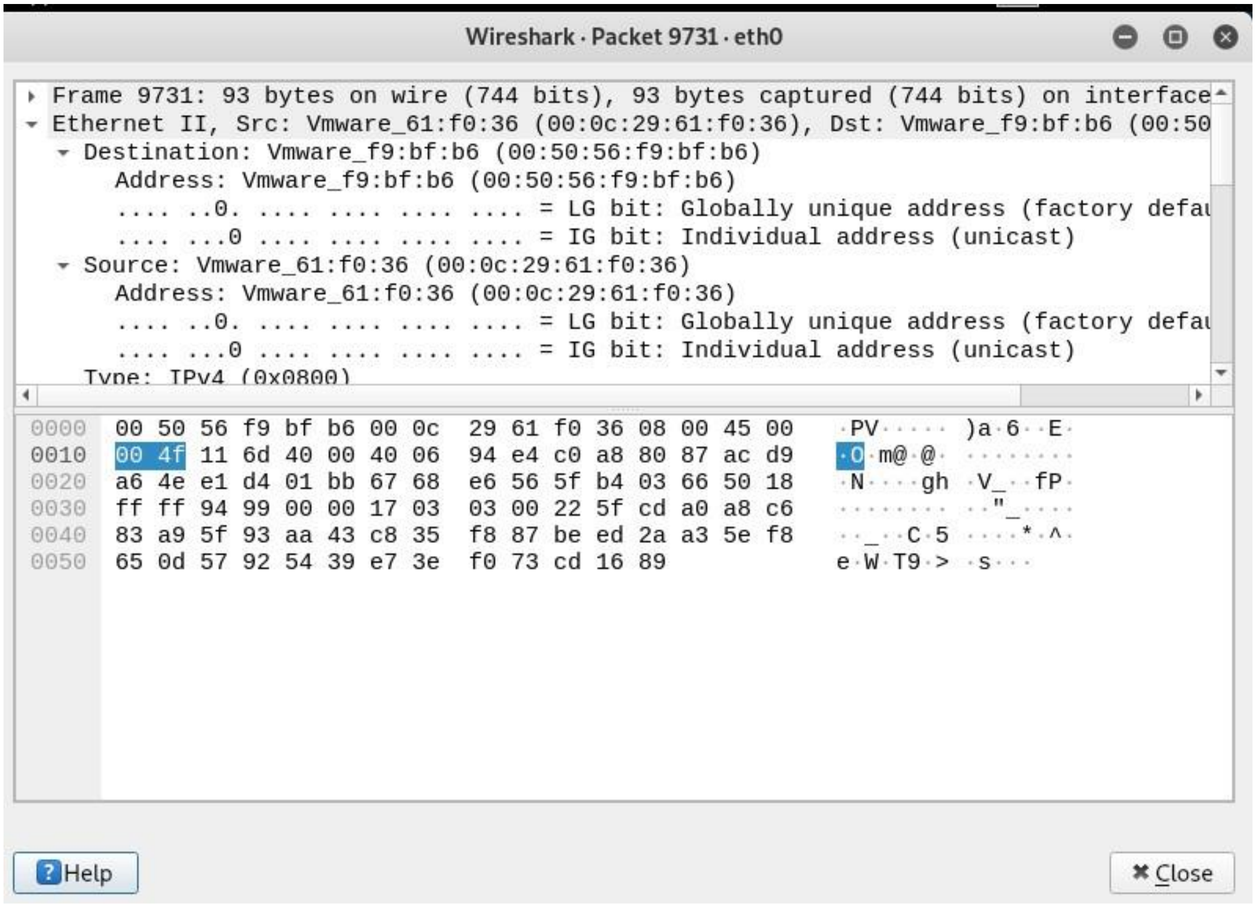
--There are 2 protocols is available udp and tcp.

--TCP stream Assembles data from protocols that utilize TCP, such as HTTP and FTP.

--UDP stream Assembles data from protocols that utilize UDP, such as DNS.

--SSL stream Assembles data from protocols that are encrypted, such as HTTPS. You must supply keys to decrypt the traffic.

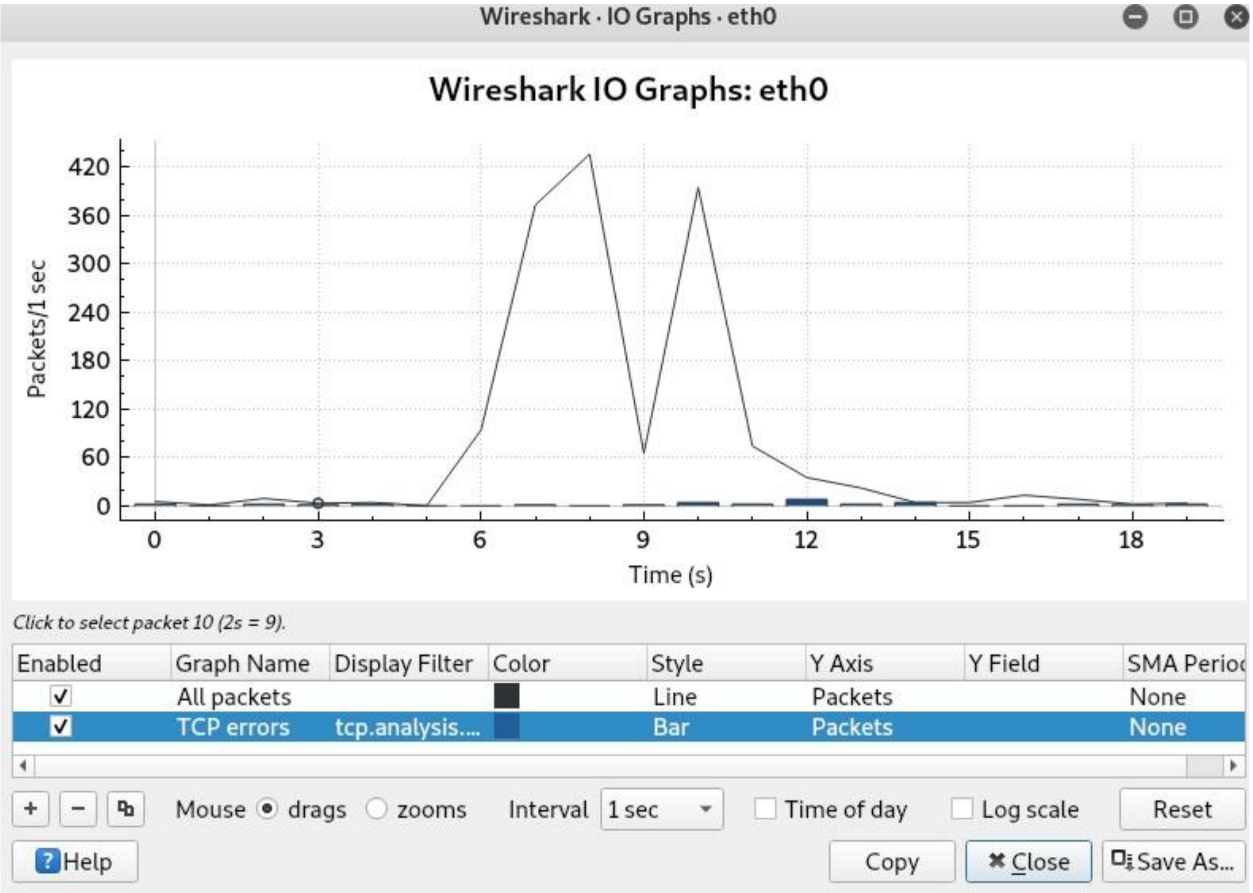
--HTTP stream Assembles and decompresses data from the HTTP protocol. This is useful when following HTTP data via TCP stream doesn't decode the HTTP payload fully



```
▸ Frame 9731: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface
▾ Ethernet II, Src: Vmware_61:f0:36 (00:0c:29:61:f0:36), Dst: Vmware_f9:bf:b6 (00:50
  ▾ Destination: Vmware_f9:bf:b6 (00:50:56:f9:bf:b6)
      Address: Vmware_f9:bf:b6 (00:50:56:f9:bf:b6)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory defau
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: Vmware_61:f0:36 (00:0c:29:61:f0:36)
      Address: Vmware_61:f0:36 (00:0c:29:61:f0:36)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory defau
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

```
0000  00 50 56 f9 bf b6 00 0c  29 61 f0 36 08 00 45 00   ·PV·····  )a·6··E·
0010  00 4f 11 6d 40 00 40 06  94 e4 c0 a8 80 87 ac d9   ·O·m@·@·  ········
0020  a6 4e e1 d4 01 bb 67 68  e6 56 5f b4 03 66 50 18   ·N····gh  ·V_··fP·
0030  ff ff 94 99 00 00 17 03  03 00 22 5f cd a0 a8 c6   ········  ··"_····
0040  83 a9 5f 93 aa 43 c8 35  f8 87 be ed 2a a3 5e f8   ··_··C·5  ····*·^·
0050  65 0d 57 92 54 39 e7 3e  f0 73 cd 16 89            e·W·T9·>  ·s···
```

13) Capture the traffic from any secure or non-secure website and observe the IO Graphs.

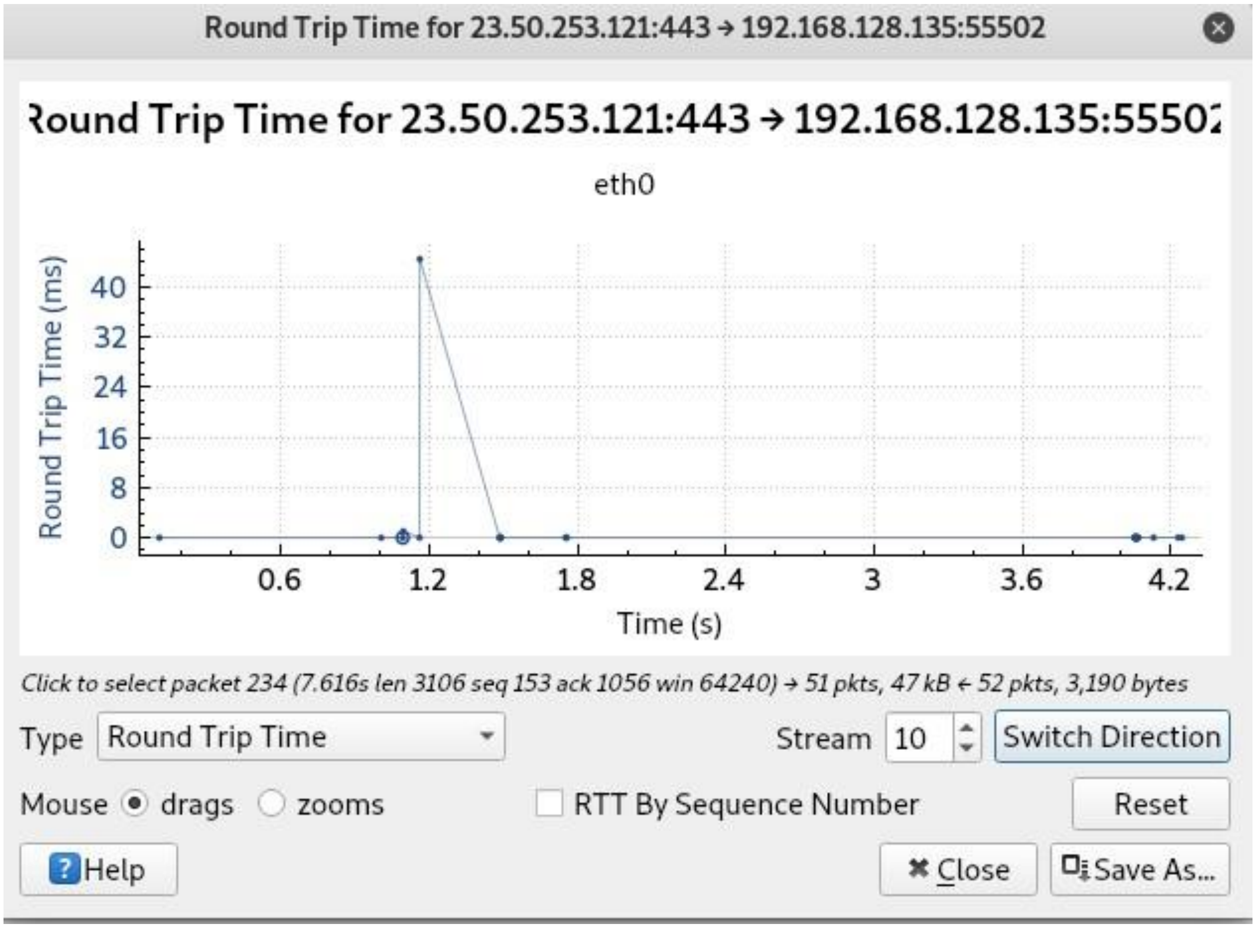


Click to select packet 10 (2s = 9).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Perio
☑	All packets		■	Line	Packets		None
☑	TCP errors	tcp.analysis....	■	Bar	Packets		None

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale Reset

14) Capture the traffic from any secure or non-secure website and observe the Round-Trip Time Graphing.


Round Trip Time for 23.50.253.121:443 → 192.168.128.135:55502
Round Trip Time for 23.50.253.121:443 → 192.168.128.135:5550
eth0
Round Trip Time (ms)
40
32
24
16
8
0
0.6
1.2
1.8
2.4
3
3.6
4.2
Time (s)
Switch Direction
Reset
Help
✖ Close
Save As...

Click to select packet 234 (7.616s len 3106 seq 153 ack 1056 win 64240) → 51 pkts, 47 kB ← 52 pkts, 3,190 bytes

Type Round Trip Time

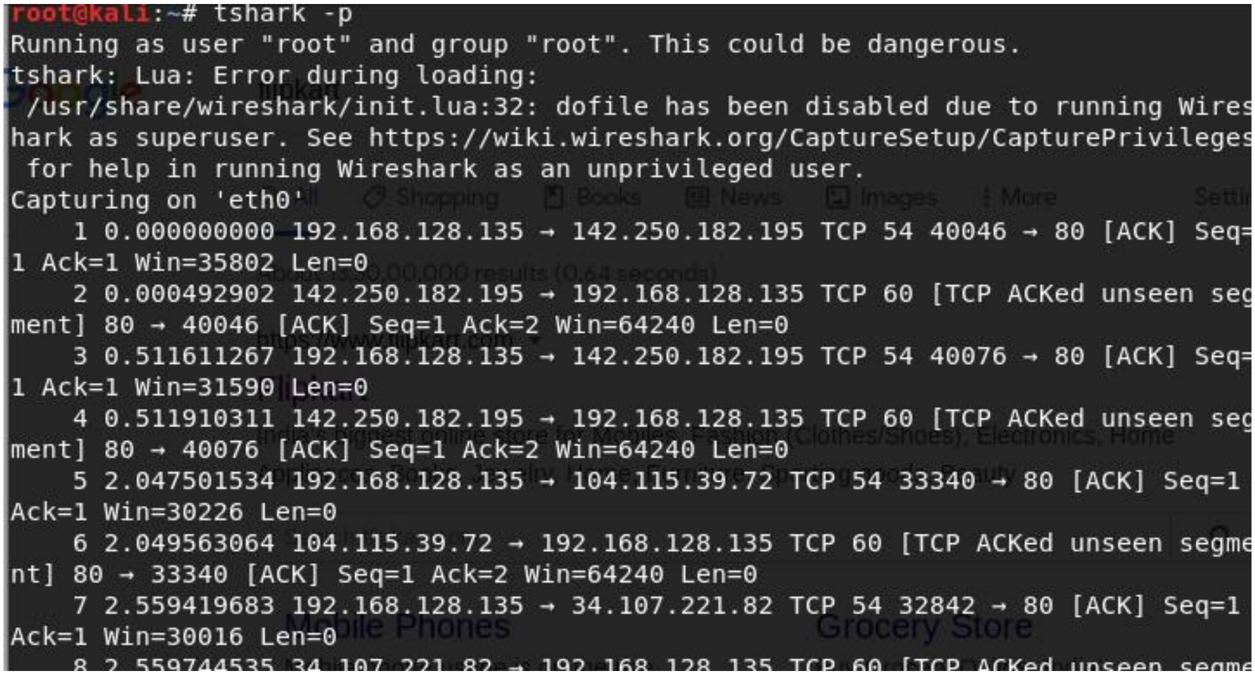
Stream 10

Mouse ● drags ○ zooms

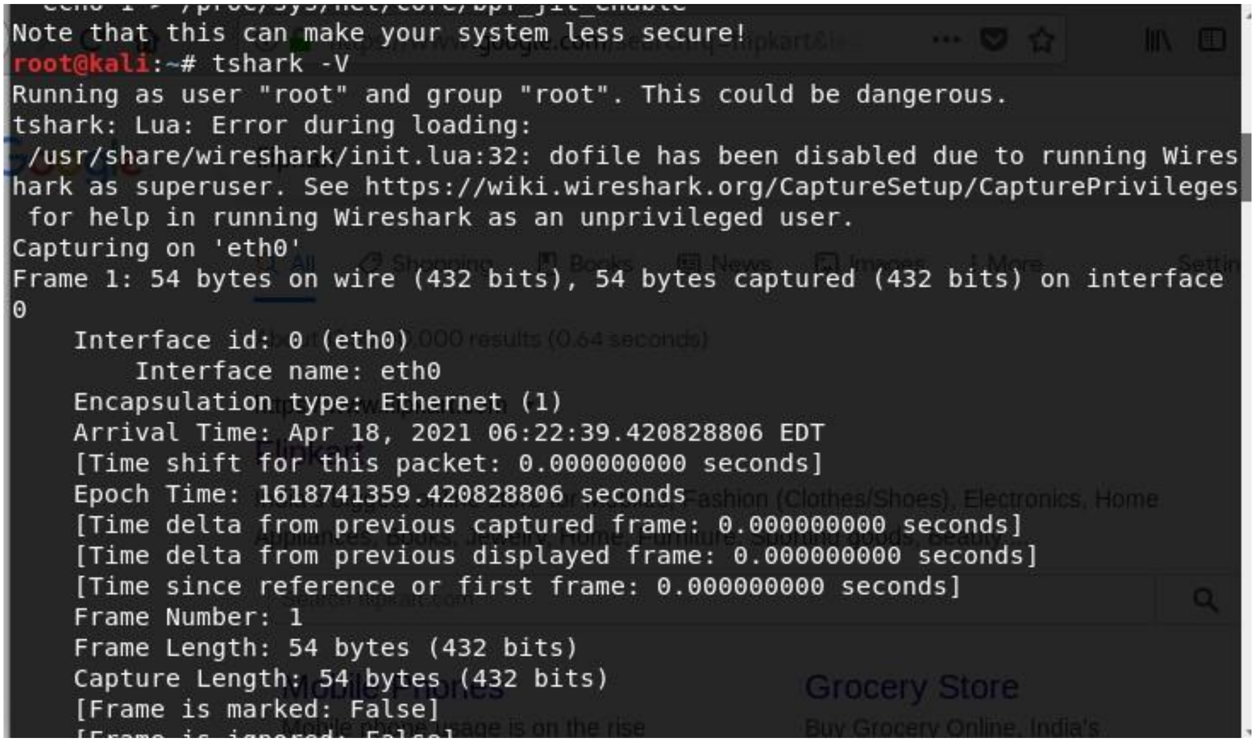
□ RTT By Sequence Number

15) What is TShark? What is tcpdump? List and take screenshot of atleast 10 commands in the Wireshark.

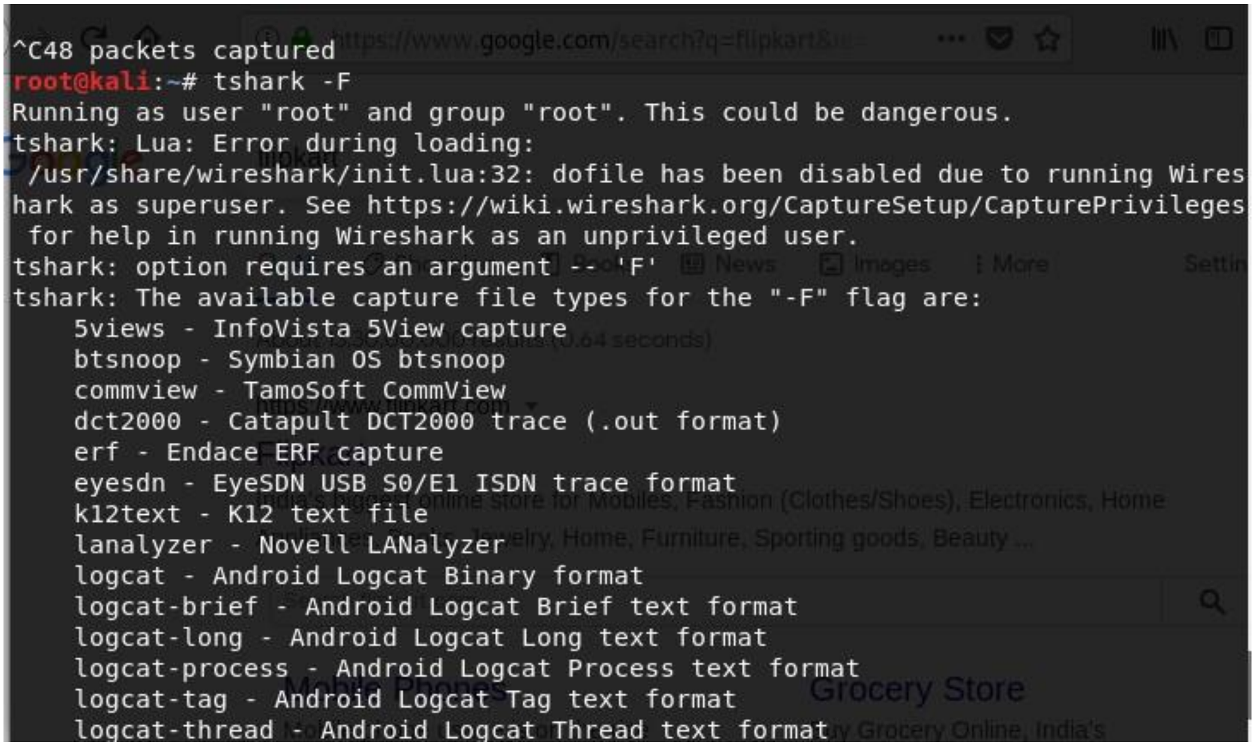
--Like Wireshark, TShark can run on multiple operating systems, but since it's not dependent on OSspecific graphics libraries, the user experience is more consistent across different OS platforms.



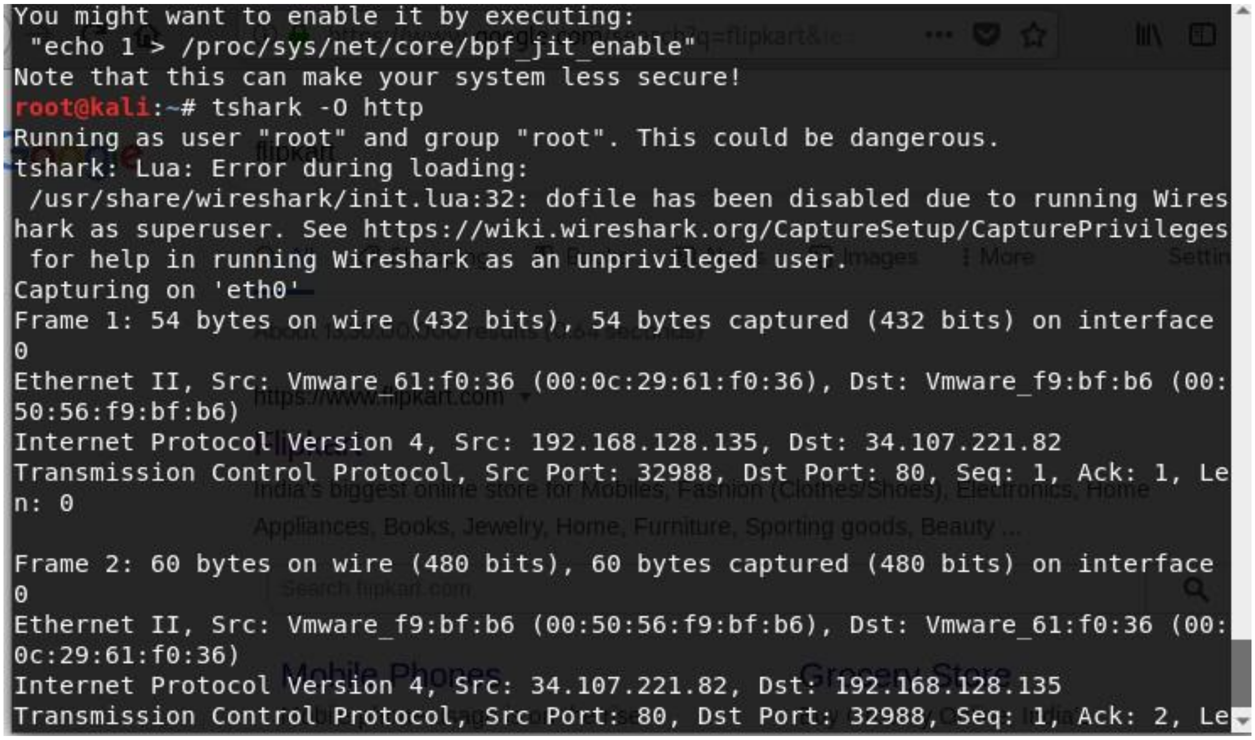
```
root@kali:~# tshark -p
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 192.168.128.135 → 142.250.182.195 TCP 54 40046 → 80 [ACK] Seq=
1 Ack=1 Win=35802 Len=0
    2 0.000492902 142.250.182.195 → 192.168.128.135 TCP 60 [TCP ACKed unseen seg
ment] 80 → 40046 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    3 0.511611267 192.168.128.135 → 142.250.182.195 TCP 54 40076 → 80 [ACK] Seq=
1 Ack=1 Win=31590 Len=0
    4 0.511910311 142.250.182.195 → 192.168.128.135 TCP 60 [TCP ACKed unseen seg
ment] 80 → 40076 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    5 2.047501534 192.168.128.135 → 104.115.39.72 TCP 54 33340 → 80 [ACK] Seq=1
Ack=1 Win=30226 Len=0
    6 2.049563064 104.115.39.72 → 192.168.128.135 TCP 60 [TCP ACKed unseen segme
nt] 80 → 33340 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    7 2.559419683 192.168.128.135 → 34.107.221.82 TCP 54 32842 → 80 [ACK] Seq=1
Ack=1 Win=30016 Len=0
    8 2.559744535 34.107.221.82 → 192.168.128.135 TCP 60 [TCP ACKed unseen segme
```



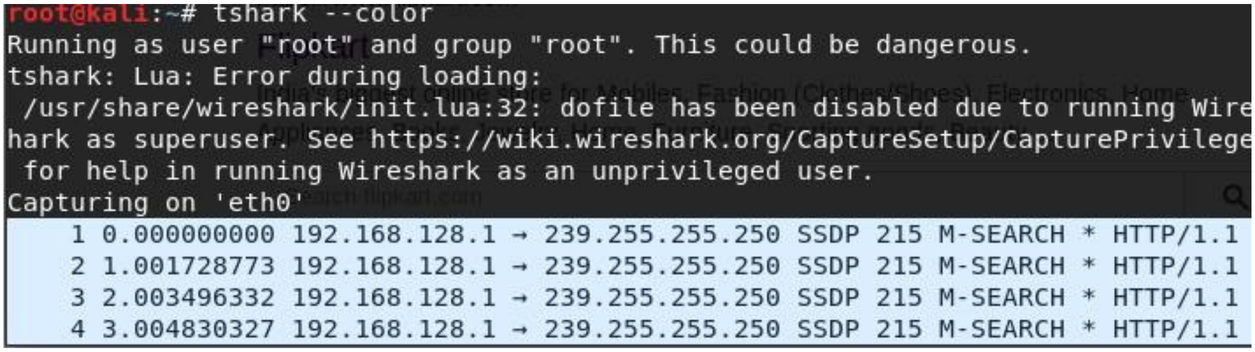
```
  echo 1 > /proc/sys/net/core/bpf_jit_enable
Note that this can make your system less secure!
root@kali:~# tshark -V
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
0
    Interface id: 0 (eth0)
        Interface name: eth0
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 18, 2021 06:22:39.420828806 EDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1618741359.420828806 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 54 bytes (432 bits)
    Capture Length: 54 bytes (432 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
```



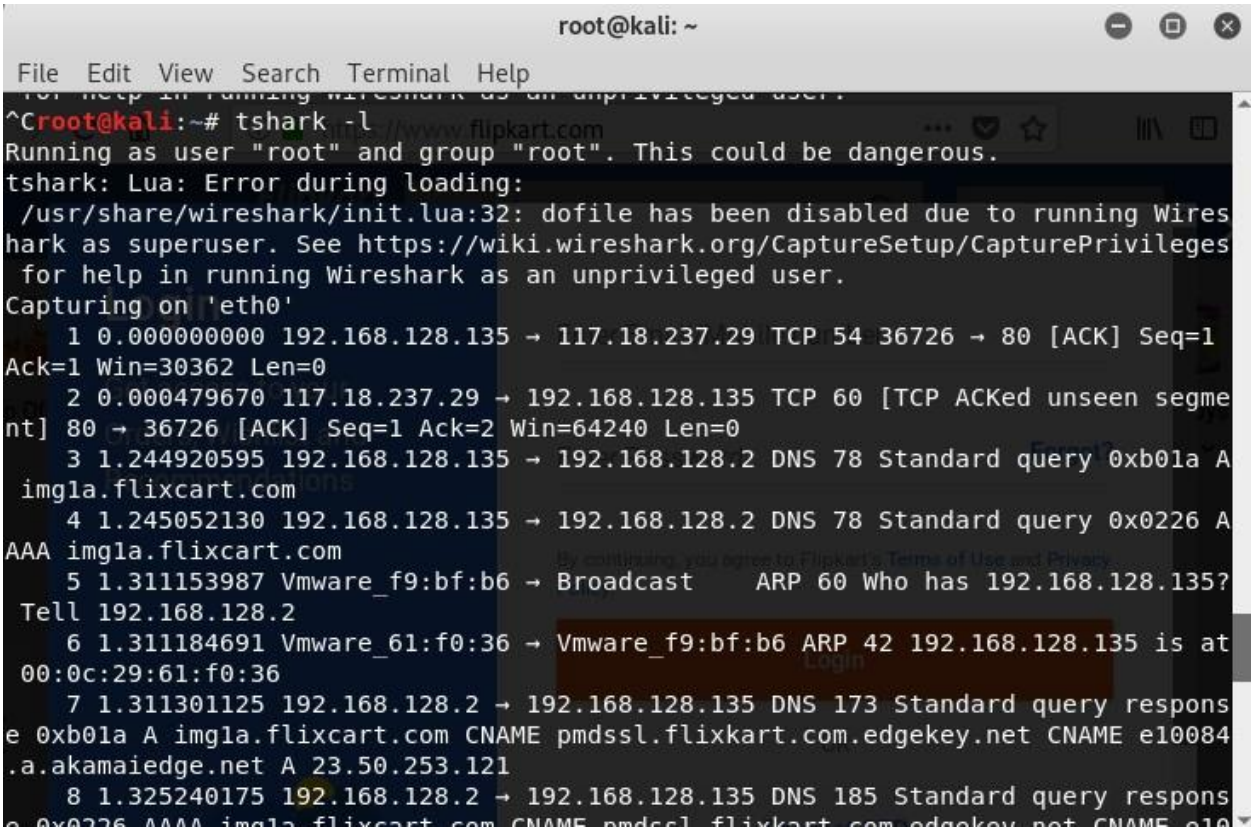
```
^C48 packets captured
root@kali:~# tshark -F
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
tshark: option requires an argument -- 'F'
tshark: The available capture file types for the "-F" flag are:
    5views - InfoVista 5View capture
    btsnoop - Symbian OS btsnoop
    commview - TamoSoft CommView
    dct2000 - Catapult DCT2000 trace (.out format)
    erf - Endace ERF capture
    eyesdn - EyeSDN USB S0/E1 ISDN trace format
    k12text - K12 text file
    lanalyzer - Novell LANalyzer
    logcat - Android Logcat Binary format
    logcat-brief - Android Logcat Brief text format
    logcat-long - Android Logcat Long text format
    logcat-process - Android Logcat Process text format
    logcat-tag - Android Logcat Tag text format
    logcat-thread - Android Logcat Thread text format
```



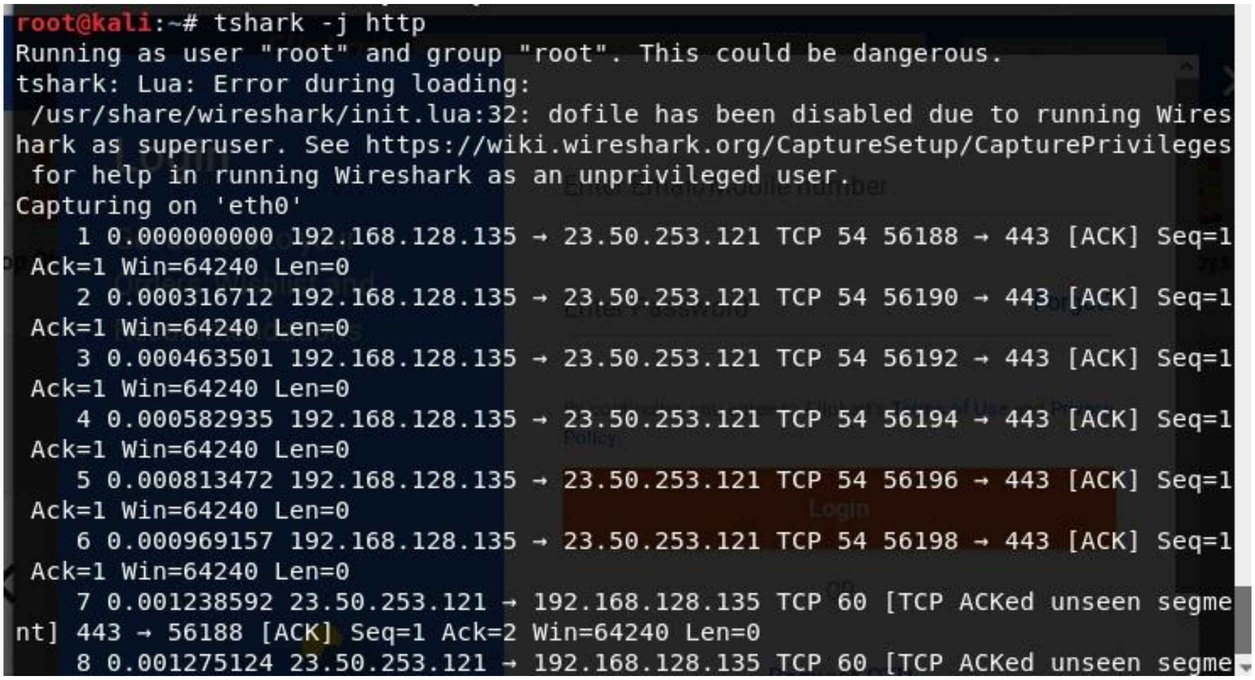
```
You might want to enable it by executing:
 "echo 1 > /proc/sys/net/core/bpf_jit_enable"
Note that this can make your system less secure!
root@kali:~# tshark -O http
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
0
Ethernet II, Src: Vmware_61:f0:36 (00:0c:29:61:f0:36), Dst: Vmware_f9:bf:b6 (00:
50:56:f9:bf:b6)
Internet Protocol Version 4, Src: 192.168.128.135, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 32988, Dst Port: 80, Seq: 1, Ack: 1, Le
n: 0

Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface
0
Ethernet II, Src: Vmware_f9:bf:b6 (00:50:56:f9:bf:b6), Dst: Vmware_61:f0:36 (00:
0c:29:61:f0:36)
Internet Protocol Version 4, Src: 34.107.221.82, Dst: 192.168.128.135
Transmission Control Protocol, Src Port: 80, Dst Port: 32988, Seq: 1, Ack: 2, Le
```



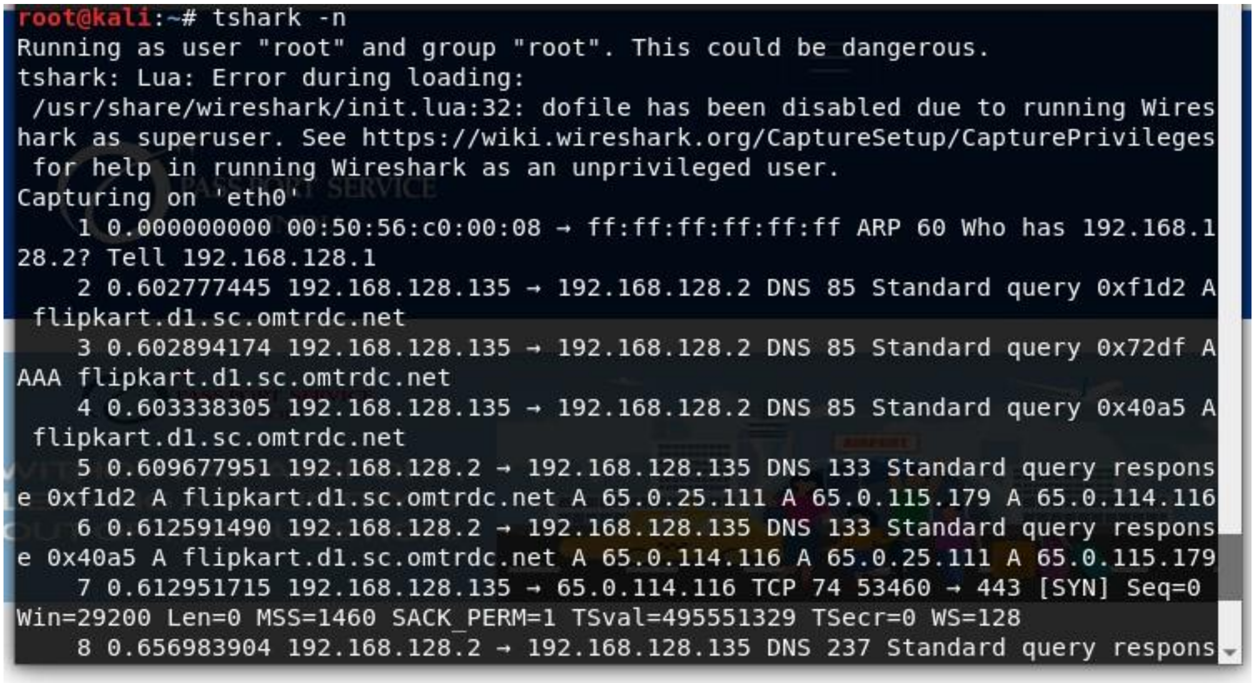
```
root@kali:~# tshark --color
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wire
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivilege
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 192.168.128.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
    2 1.001728773 192.168.128.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
    3 2.003496332 192.168.128.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
    4 3.004830327 192.168.128.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
```



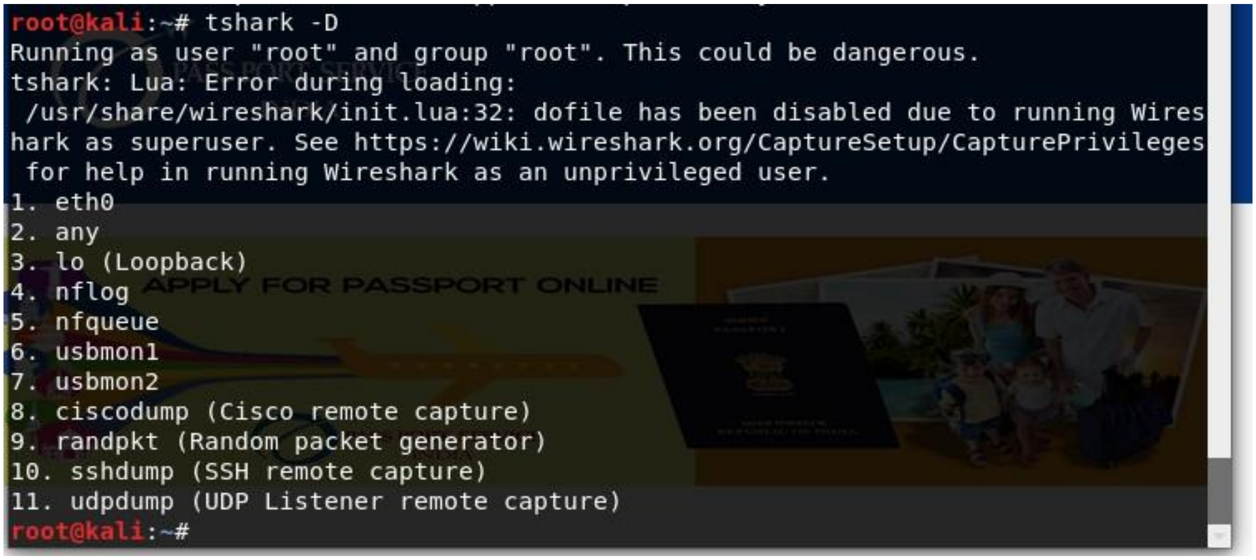
```
 for help in running Wireshark as an unprivileged user.
^Croot@kali:~# tshark -l
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 192.168.128.135 → 117.18.237.29 TCP 54 36726 → 80 [ACK] Seq=1
Ack=1 Win=30362 Len=0
    2 0.000479670 117.18.237.29 → 192.168.128.135 TCP 60 [TCP ACKed unseen segme
nt] 80 → 36726 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    3 1.244920595 192.168.128.135 → 192.168.128.2 DNS 78 Standard query 0xb01a A
 img1a.flixcart.com
    4 1.245052130 192.168.128.135 → 192.168.128.2 DNS 78 Standard query 0x0226 A
AAA img1a.flixcart.com
    5 1.311153987 Vmware_f9:bf:b6 → Broadcast    ARP 60 Who has 192.168.128.135?
 Tell 192.168.128.2
    6 1.311184691 Vmware_61:f0:36 → Vmware_f9:bf:b6 ARP 42 192.168.128.135 is at
 00:0c:29:61:f0:36
    7 1.311301125 192.168.128.2 → 192.168.128.135 DNS 173 Standard query respons
e 0xb01a A img1a.flixcart.com CNAME pmdssl.flixkart.com.edgekey.net CNAME e10084
.a.akamaiedge.net A 23.50.253.121
    8 1.325240175 192.168.128.2 → 192.168.128.135 DNS 185 Standard query respons
e 0x0226 AAAA img1a.flixcart.com CNAME pmdssl.flixkart.com.edgekey.net CNAME e10
```



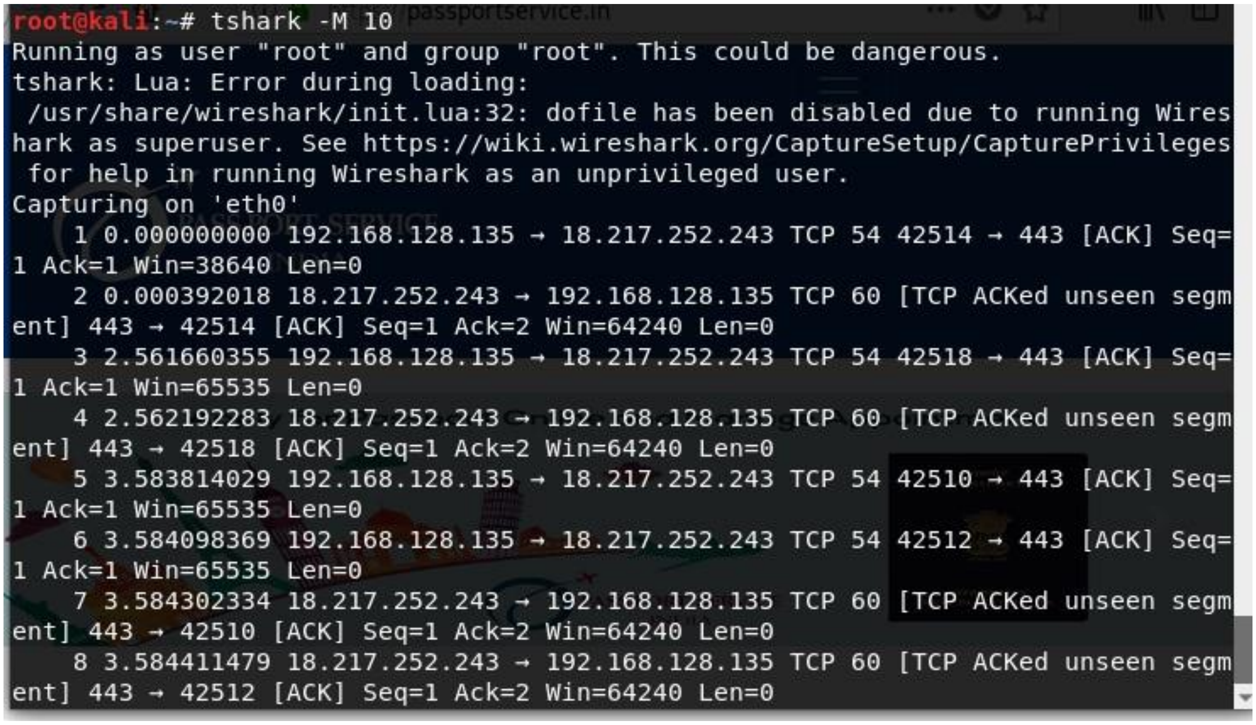
```
root@kali:~# tshark -j http
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 192.168.128.135 → 23.50.253.121 TCP 54 56188 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    2 0.000316712 192.168.128.135 → 23.50.253.121 TCP 54 56190 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    3 0.000463501 192.168.128.135 → 23.50.253.121 TCP 54 56192 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    4 0.000582935 192.168.128.135 → 23.50.253.121 TCP 54 56194 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    5 0.000813472 192.168.128.135 → 23.50.253.121 TCP 54 56196 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    6 0.000969157 192.168.128.135 → 23.50.253.121 TCP 54 56198 → 443 [ACK] Seq=1
 Ack=1 Win=64240 Len=0
    7 0.001238592 23.50.253.121 → 192.168.128.135 TCP 60 [TCP ACKed unseen segme
nt] 443 → 56188 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    8 0.001275124 23.50.253.121 → 192.168.128.135 TCP 60 [TCP ACKed unseen segme
```



```
root@kali:~# tshark -n
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 00:50:56:c0:00:08 → ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.1
28.2? Tell 192.168.128.1
    2 0.602777445 192.168.128.135 → 192.168.128.2 DNS 85 Standard query 0xf1d2 A
 flipkart.d1.sc.omtrdc.net
    3 0.602894174 192.168.128.135 → 192.168.128.2 DNS 85 Standard query 0x72df A
AAA flipkart.d1.sc.omtrdc.net
    4 0.603338305 192.168.128.135 → 192.168.128.2 DNS 85 Standard query 0x40a5 A
 flipkart.d1.sc.omtrdc.net
    5 0.609677951 192.168.128.2 → 192.168.128.135 DNS 133 Standard query respons
e 0xf1d2 A flipkart.d1.sc.omtrdc.net A 65.0.25.111 A 65.0.115.179 A 65.0.114.116
    6 0.612591490 192.168.128.2 → 192.168.128.135 DNS 133 Standard query respons
e 0x40a5 A flipkart.d1.sc.omtrdc.net A 65.0.114.116 A 65.0.25.111 A 65.0.115.179
    7 0.612951715 192.168.128.135 → 65.0.114.116 TCP 74 53460 → 443 [SYN] Seq=0
Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=495551329 TSecr=0 WS=128
    8 0.656983904 192.168.128.2 → 192.168.128.135 DNS 237 Standard query respons
```



```
root@kali:~# tshark -D
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
1. eth0
2. any
3. lo (Loopback)
4. nflog
5. nfqueue
6. usbmon1
7. usbmon2
8. ciscodump (Cisco remote capture)
9. randpkt (Random packet generator)
10. sshdump (SSH remote capture)
11. udpdump (UDP Listener remote capture)
root@kali:~#
```



```
root@kali:~# tshark -M 10
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
 /usr/share/wireshark/init.lua:32: dofile has been disabled due to running Wires
hark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges
 for help in running Wireshark as an unprivileged user.
Capturing on 'eth0'
    1 0.000000000 192.168.128.135 → 18.217.252.243 TCP 54 42514 → 443 [ACK] Seq=
1 Ack=1 Win=38640 Len=0
    2 0.000392018 18.217.252.243 → 192.168.128.135 TCP 60 [TCP ACKed unseen segm
ent] 443 → 42514 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    3 2.561660355 192.168.128.135 → 18.217.252.243 TCP 54 42518 → 443 [ACK] Seq=
1 Ack=1 Win=65535 Len=0
    4 2.562192283 18.217.252.243 → 192.168.128.135 TCP 60 [TCP ACKed unseen segm
ent] 443 → 42518 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    5 3.583814029 192.168.128.135 → 18.217.252.243 TCP 54 42510 → 443 [ACK] Seq=
1 Ack=1 Win=65535 Len=0
    6 3.584098369 192.168.128.135 → 18.217.252.243 TCP 54 42512 → 443 [ACK] Seq=
1 Ack=1 Win=65535 Len=0
    7 3.584302334 18.217.252.243 → 192.168.128.135 TCP 60 [TCP ACKed unseen segm
ent] 443 → 42510 [ACK] Seq=1 Ack=2 Win=64240 Len=0
    8 3.584411479 18.217.252.243 → 192.168.128.135 TCP 60 [TCP ACKed unseen segm
ent] 443 → 42512 [ACK] Seq=1 Ack=2 Win=64240 Len=0
```