Experiment No. – 2				
Date of Performance:				
Date of Submission:				
Program Execution/ formation/ correction/ ethical practices (06)	Timely Submission (01)	Viva (03)	Experiment Total (10)	Sign with Date

Experiment No. 2
Packet sniffer tools in Wireshark.

2.1Aim: Use Packet sniffing tool: Wireshark to understand the operation of TCP/IP layers.

2.2Course Outcome: Explain the need for Cyber Security and its aspects.

2.3 Learning Objectives: Using Wireshark tool to explore networking algorithms and protocols.

2.4 Requirement: Kali Linux

2.5 Related Theory:

Wireshark-
Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

Packet Capture: Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.

Filtering: Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.

Visualization: Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

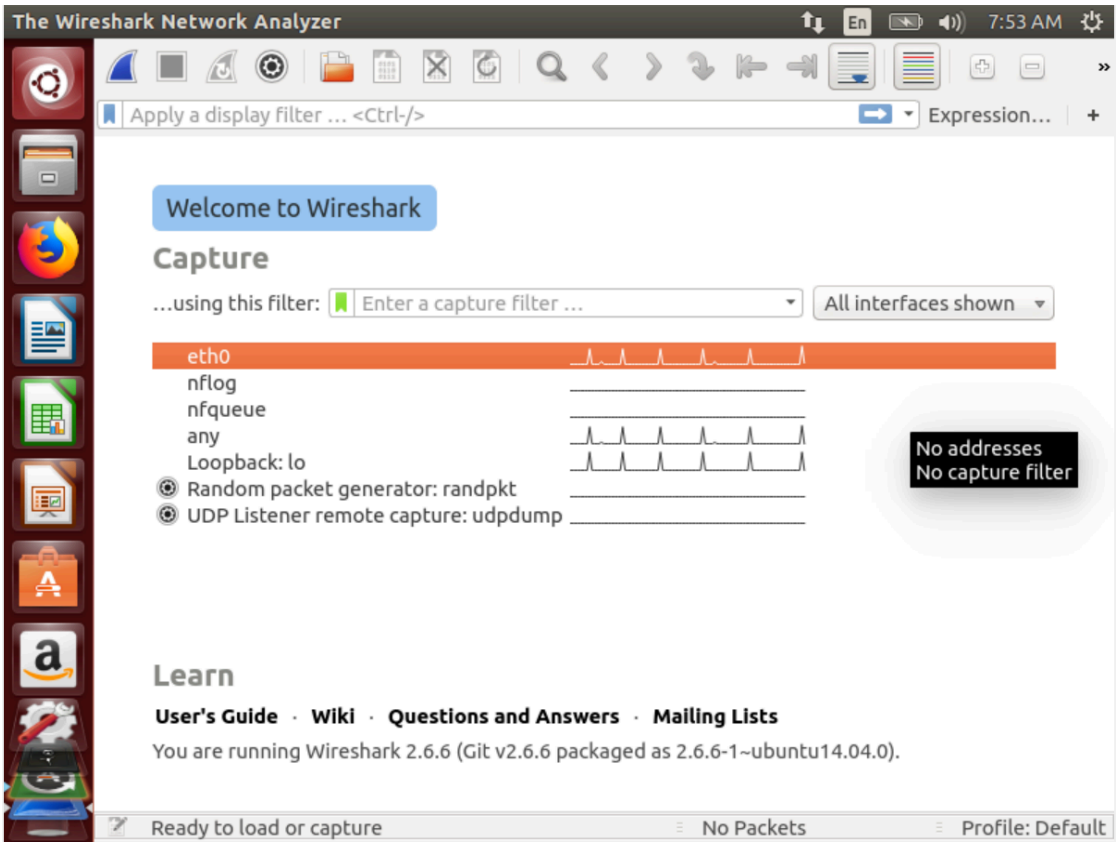


Figure 2.1 Wireshark

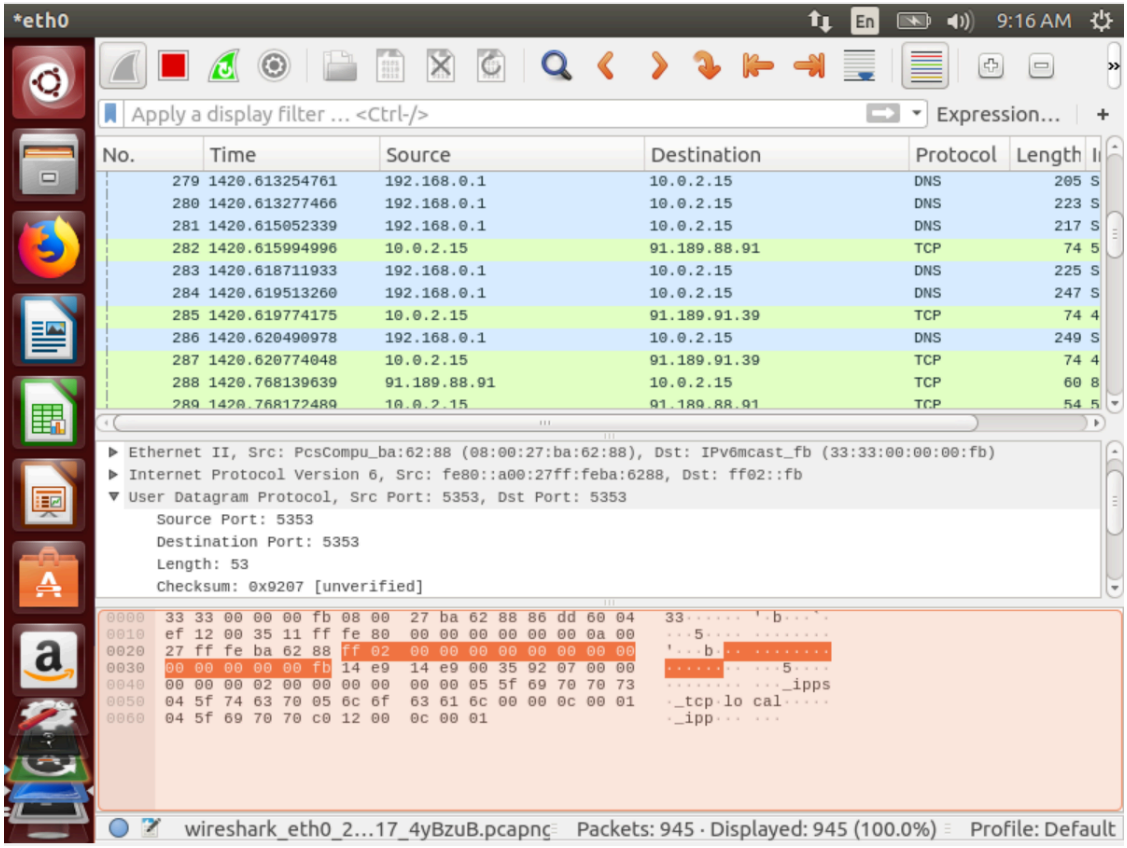


Figure 2.2 Viewing a packet capture in Wireshark

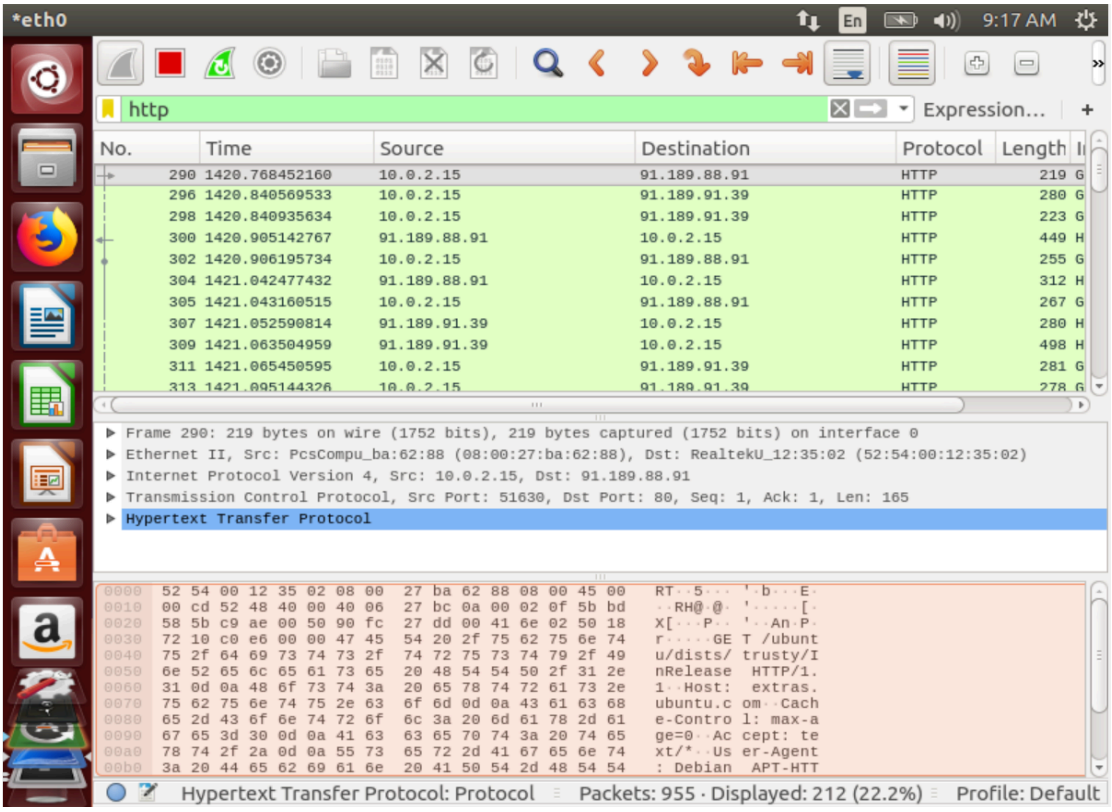


Figure 2.3 Drilling down into a packet to identify a network problem using Wireshark

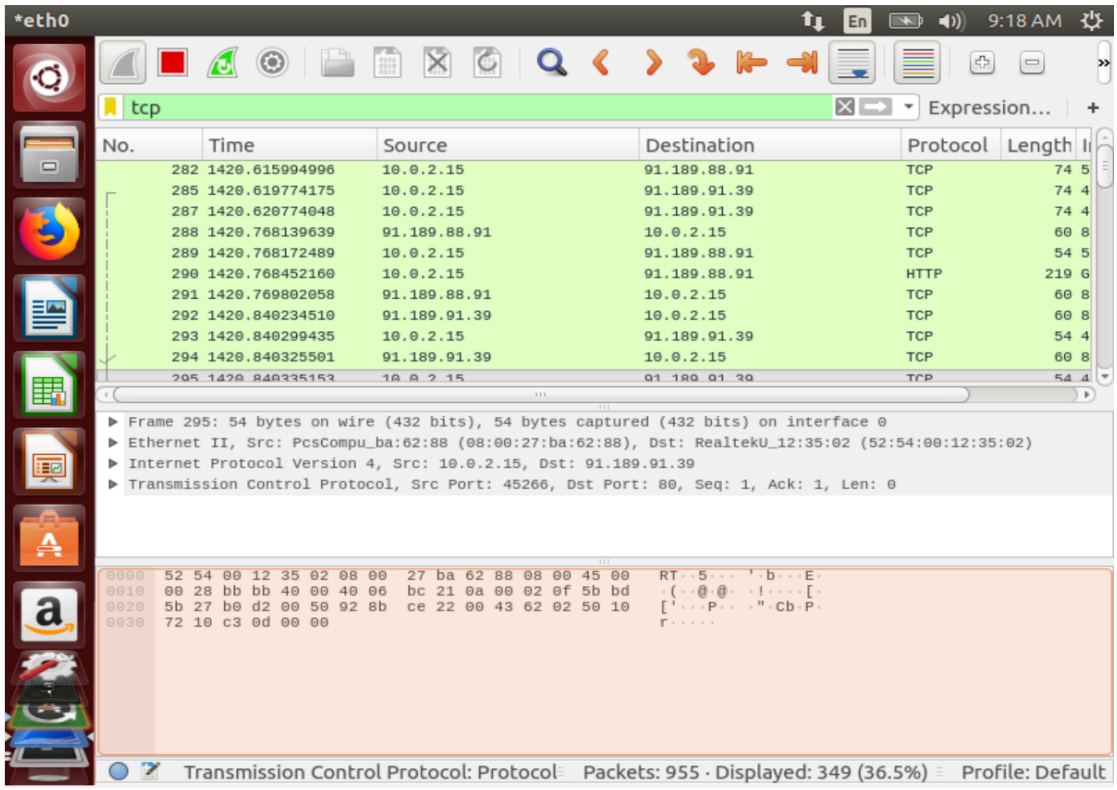


Figure 2.4 Drilling down into a packet to identify a network problem using Wireshark

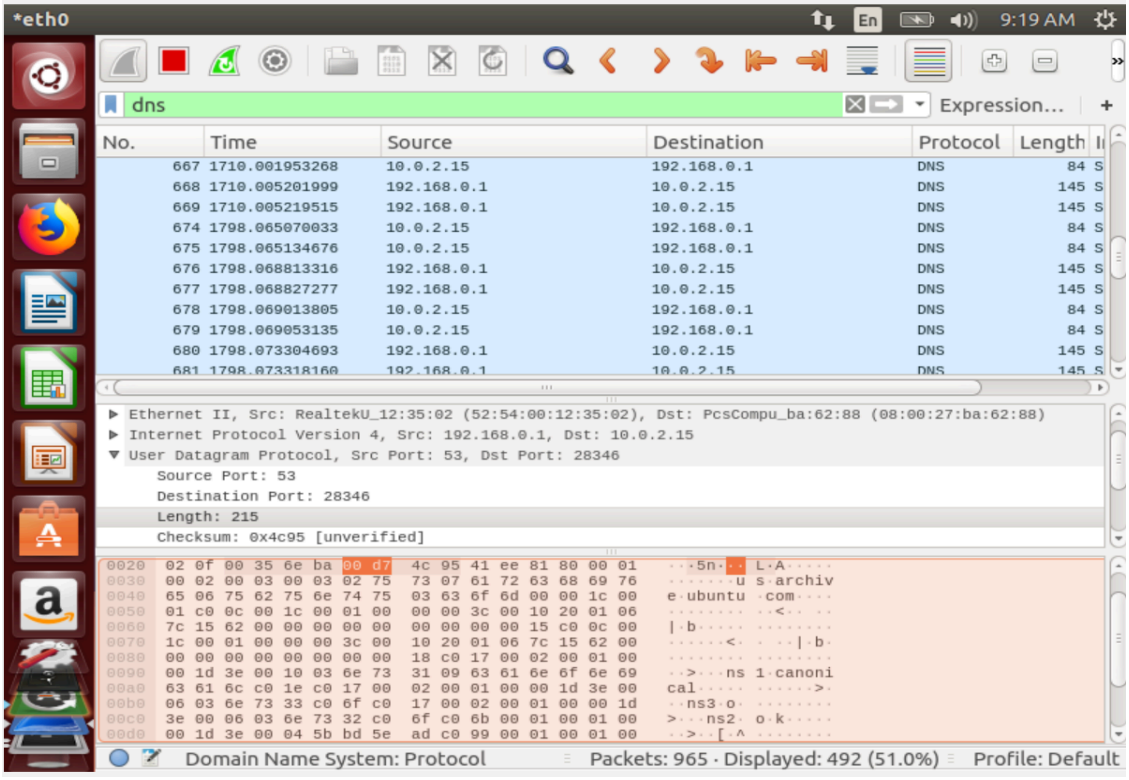
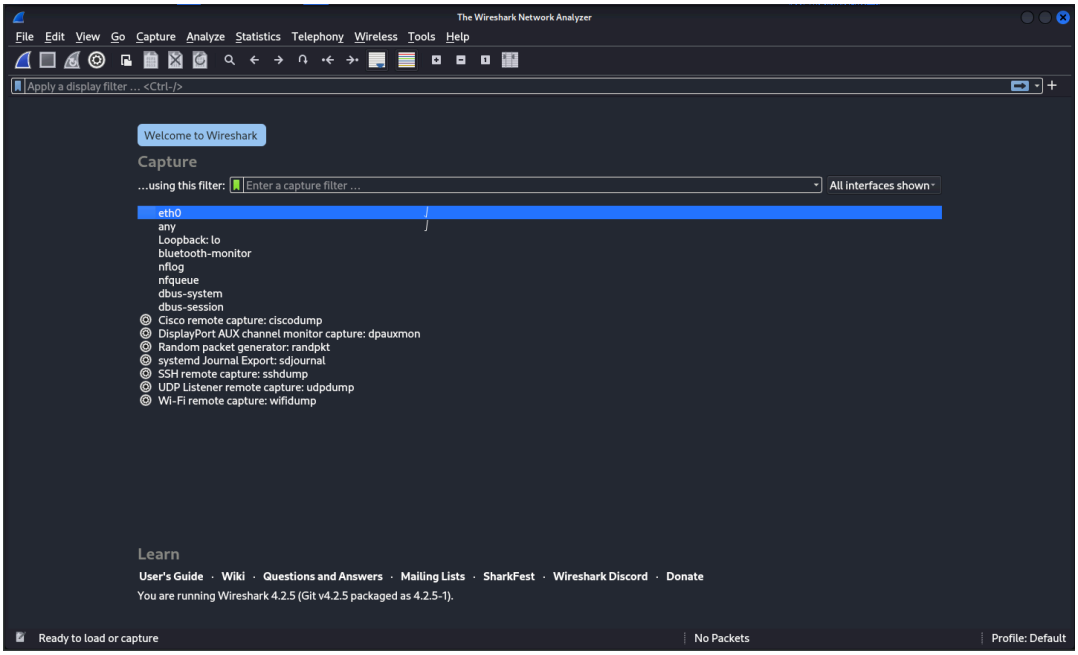
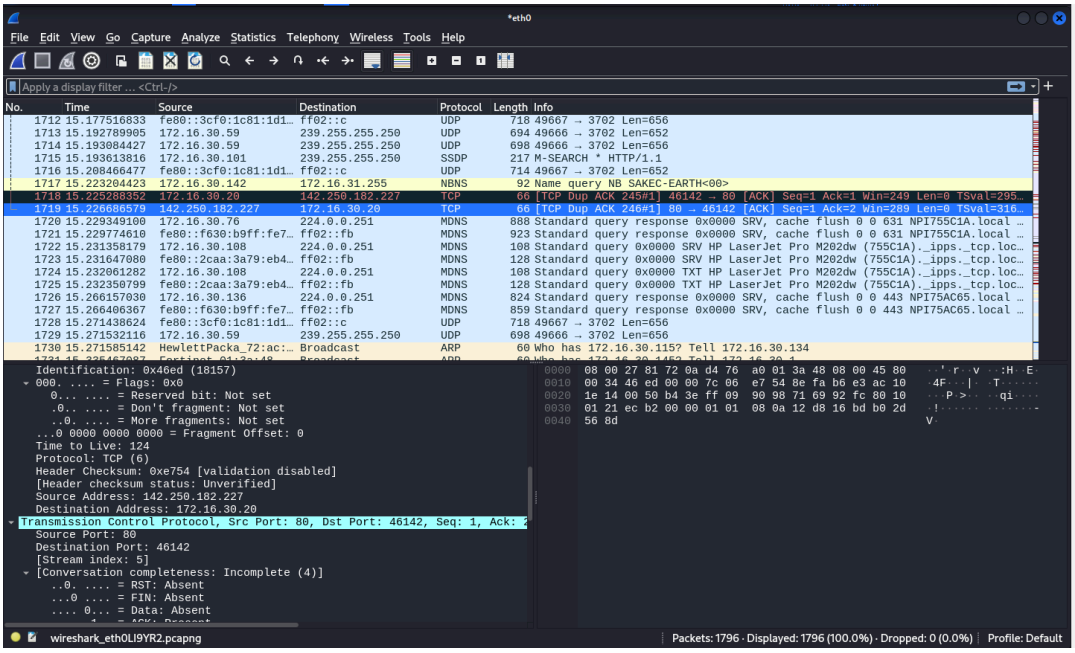
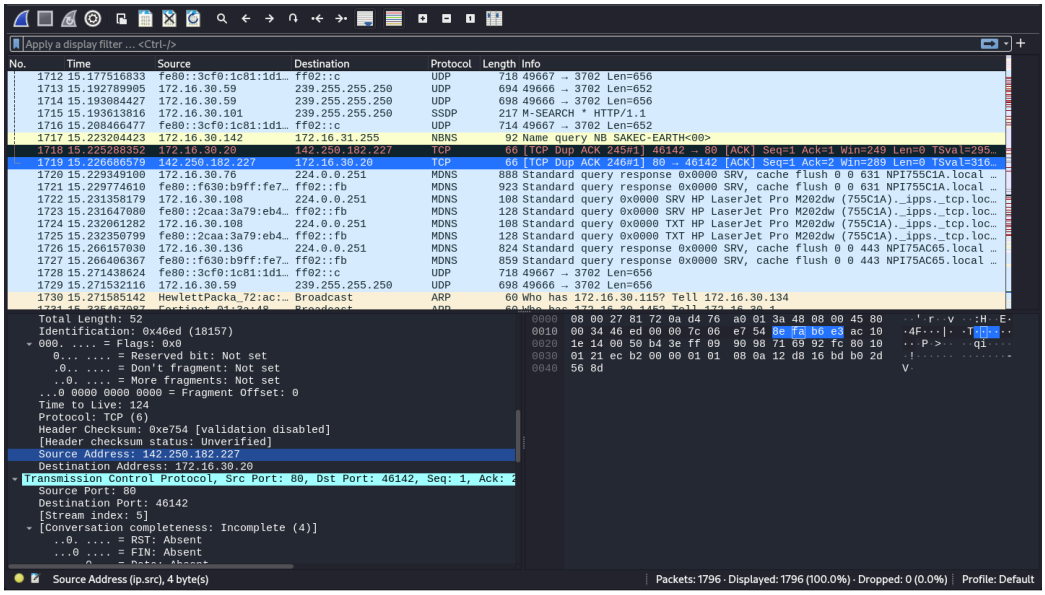


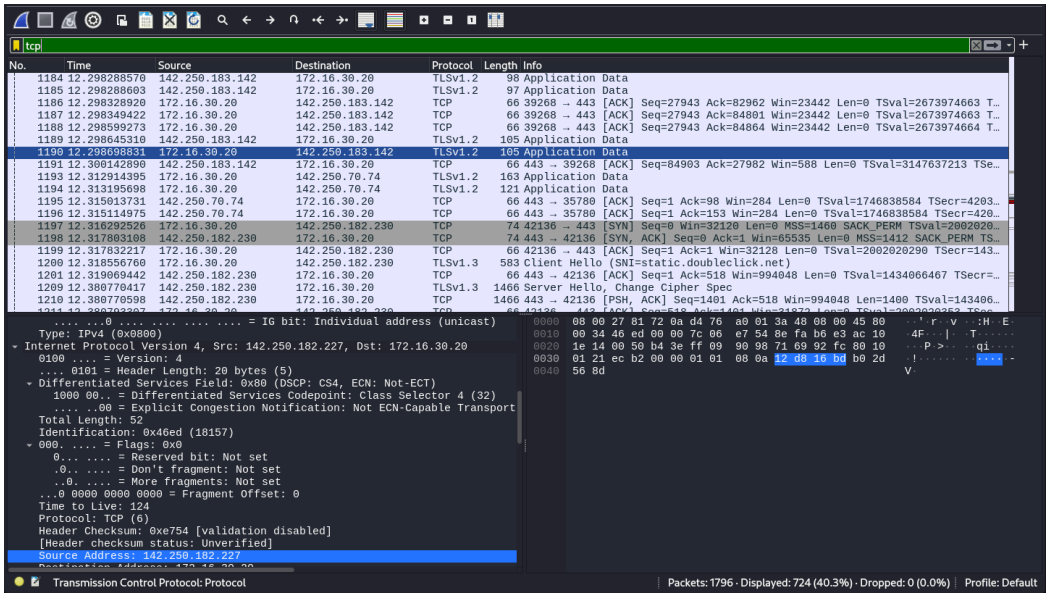
Figure 2.5 Applying filter

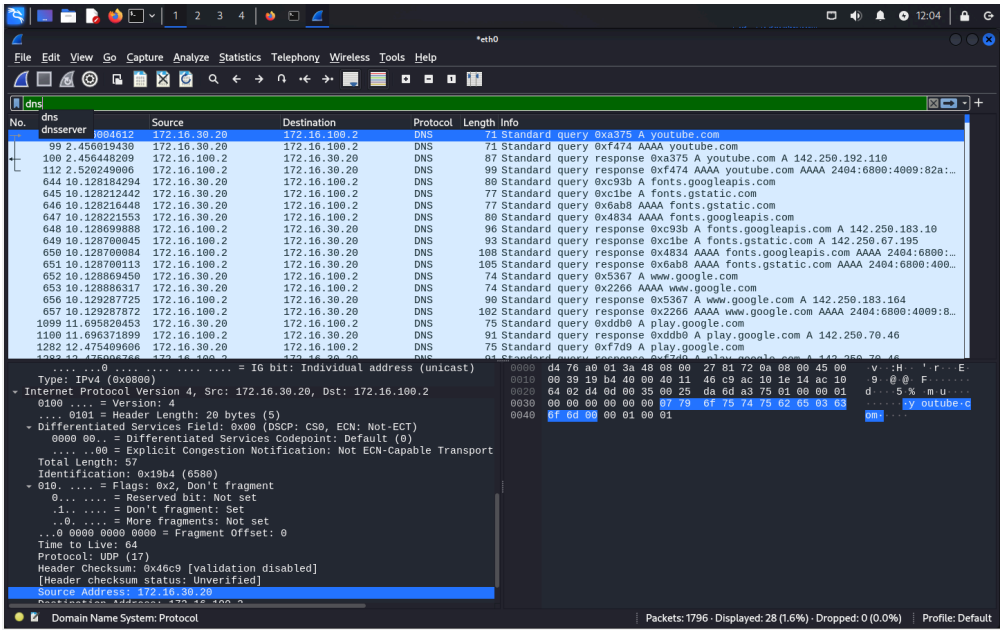
2.6 Simulated Output:






Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1712 15.177516833 fe80::3cf0:1c81:1d1… ff02::c UDP 718 49667 → 3702 Len=656
1713 15.192789905 172.16.30.59 239.255.255.250 UDP 694 49666 → 3702 Len=652
1714 15.193084427 172.16.30.59 239.255.255.250 UDP 698 49666 → 3702 Len=656
1715 15.193613816 172.16.30.101 239.255.255.250 SSDP 217 M-SEARCH * HTTP/1.1
1716 15.208466477 fe80::3cf0:1c81:1d1… ff02::c UDP 714 49667 → 3702 Len=652
1717 15.223204423 172.16.30.142 172.16.31.255 NBNS 92 Name query NB SAKEC-EARTH<00>
1718 15.225288352 172.16.30.20 142.250.182.227 TCP 66 [TCP Dup ACK 245#1] 46142 → 80 [ACK] Seq=1 Ack=1 Win=249 Len=0 TSval=295…
1719 15.226686579 142.250.182.227 172.16.30.20 TCP 66 [TCP Dup ACK 246#1] 80 → 46142 [ACK] Seq=1 Ack=2 Win=289 Len=0 TSval=316…
1720 15.229349100 172.16.30.76 224.0.0.251 MDNS 888 Standard query response 0x0000 SRV, cache flush 0 0 631 NPI755C1A.local …
1721 15.229774610 fe80::f630:b9ff:fe7… ff02::fb MDNS 923 Standard query response 0x0000 SRV, cache flush 0 0 631 NPI755C1A.local …
1722 15.231358179 172.16.30.108 224.0.0.251 MDNS 108 Standard query 0x0000 SRV HP LaserJet Pro M202dw (755C1A)._ipps._tcp.loc…
1723 15.231647080 fe80::2caa:3a79:eb4… ff02::fb MDNS 128 Standard query 0x0000 SRV HP LaserJet Pro M202dw (755C1A)._ipps._tcp.loc…
1724 15.232061282 172.16.30.108 224.0.0.251 MDNS 108 Standard query 0x0000 TXT HP LaserJet Pro M202dw (755C1A)._ipps._tcp.loc…
1725 15.232350799 fe80::2caa:3a79:eb4… ff02::fb MDNS 128 Standard query 0x0000 TXT HP LaserJet Pro M202dw (755C1A)._ipps._tcp.loc…
1726 15.266157030 172.16.30.136 224.0.0.251 MDNS 824 Standard query response 0x0000 SRV, cache flush 0 0 443 NPI75AC65.local …
1727 15.266406367 fe80::f630:b9ff:fe7… ff02::fb MDNS 859 Standard query response 0x0000 SRV, cache flush 0 0 443 NPI75AC65.local …
1728 15.271438624 fe80::3cf0:1c81:1d1… ff02::c UDP 718 49667 → 3702 Len=656
1729 15.271532116 172.16.30.59 239.255.255.250 UDP 698 49666 → 3702 Len=656
1730 15.271585142 HewlettPacka_72:ac:… Broadcast ARP 60 Who has 172.16.30.115? Tell 172.16.30.134
Total Length: 52
Identification: 0x46ed (18157)
000. .... = Flags: 0x0
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 124
Protocol: TCP (6)
Header Checksum: 0xe754 [validation disabled]
[Header checksum status: Unverified]
Source Address: 142.250.182.227
Destination Address: 172.16.30.20
Transmission Control Protocol, Src Port: 80, Dst Port: 46142, Seq: 1, Ack: 2
Source Port: 80
Destination Port: 46142
[Stream index: 5]
[Conversation completeness: Incomplete (4)]
..0. .... = RST: Absent
...0 .... = FIN: Absent
0000 08 00 27 81 72 0a d4 76 a0 01 3a 48 08 00 45 80
0010 00 34 46 ed 00 00 7c 06 e7 54 8e fa b6 e3 ac 10
0020 1e 14 00 50 b4 3e ff 09 90 98 71 69 92 fc 80 10
0030 01 21 ec b2 00 00 01 01 08 0a 12 d8 16 bd b0 2d
0040 56 8d
Source Address (ip.src), 4 byte(s)
Packets: 1796 · Displayed: 1796 (100.0%) · Dropped: 0 (0.0%)
Profile: Default


tcp
No. Time Source Destination Protocol Length Info
1184 12.298288570 142.250.183.142 172.16.30.20 TLSv1.2 98 Application Data
1185 12.298288603 142.250.183.142 172.16.30.20 TLSv1.2 97 Application Data
1186 12.298328920 172.16.30.20 142.250.183.142 TCP 66 39268 → 443 [ACK] Seq=27943 Ack=82962 Win=23442 Len=0 TSval=2673974663 T…
1187 12.298349422 172.16.30.20 142.250.183.142 TCP 66 39268 → 443 [ACK] Seq=27943 Ack=84801 Win=23442 Len=0 TSval=2673974663 T…
1188 12.298599273 172.16.30.20 142.250.183.142 TCP 66 39268 → 443 [ACK] Seq=27943 Ack=84864 Win=23442 Len=0 TSval=2673974664 T…
1189 12.298645310 142.250.183.142 172.16.30.20 TLSv1.2 105 Application Data
1190 12.298698831 172.16.30.20 142.250.183.142 TLSv1.2 105 Application Data
1191 12.300142890 142.250.183.142 172.16.30.20 TCP 66 443 → 39268 [ACK] Seq=84903 Ack=27982 Win=588 Len=0 TSval=3147637213 TSe…
1193 12.312914395 172.16.30.20 142.250.70.74 TLSv1.2 163 Application Data
1194 12.313195698 172.16.30.20 142.250.70.74 TLSv1.2 121 Application Data
1195 12.315013731 142.250.70.74 172.16.30.20 TCP 66 443 → 35780 [ACK] Seq=1 Ack=98 Win=284 Len=0 TSval=1746838584 TSecr=4203…
1196 12.315114975 142.250.70.74 172.16.30.20 TCP 66 443 → 35780 [ACK] Seq=1 Ack=153 Win=284 Len=0 TSval=1746838584 TSecr=420…
1197 12.316292526 172.16.30.20 142.250.182.230 TCP 74 42136 → 443 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2002020…
1198 12.317803108 142.250.182.230 172.16.30.20 TCP 74 443 → 42136 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TS…
1199 12.317832217 172.16.30.20 142.250.182.230 TCP 66 42136 → 443 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2002020290 TSecr=143…
1200 12.318556760 172.16.30.20 142.250.182.230 TLSv1.3 583 Client Hello (SNI=static.doubleclick.net)
1201 12.319069442 142.250.182.230 172.16.30.20 TCP 66 443 → 42136 [ACK] Seq=1 Ack=518 Win=994048 Len=0 TSval=1434066467 TSecr=…
1209 12.380770417 142.250.182.230 172.16.30.20 TLSv1.3 1466 Server Hello, Change Cipher Spec
1210 12.380770598 142.250.182.230 172.16.30.20 TCP 1466 443 → 42136 [PSH, ACK] Seq=1401 Ack=518 Win=994048 Len=1400 TSval=143406…
....0 .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 142.250.182.227, Dst: 172.16.30.20
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x80 (DSCP: CS4, ECN: Not-ECT)
1000 00.. = Differentiated Services Codepoint: Class Selector 4 (32)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport
Total Length: 52
Identification: 0x46ed (18157)
000. .... = Flags: 0x0
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 124
Protocol: TCP (6)
Header Checksum: 0xe754 [validation disabled]
[Header checksum status: Unverified]
Source Address: 142.250.182.227
0000 08 00 27 81 72 0a d4 76 a0 01 3a 48 08 00 45 80
0010 00 34 46 ed 00 00 7c 06 e7 54 8e fa b6 e3 ac 10
0020 1e 14 00 50 b4 3e ff 09 90 98 71 69 92 fc 80 10
0030 01 21 ec b2 00 00 01 01 08 0a 12 d8 16 bd b0 2d
0040 56 8d
Transmission Control Protocol: Protocol
Packets: 1796 · Displayed: 724 (40.3%) · Dropped: 0 (0.0%)
Profile: Default



2.7 Conclusion:

Hence we learned about Packet sniffer tools in Wireshark

2.8 Questions:

1. Wireshark used to be known as **Ethereal**
2. **http.request** Wireshark filter can be used to check all incoming requests to a HTTP Web server.
3. **ip.src == [specific IP]** Wireshark filter can be used to monitor outgoing packets from a specific system on the network.