

Google Cyber Security

3rd Exercise

Wireshark & Nmap

Full Name: Klajdi Cami

Wireshark

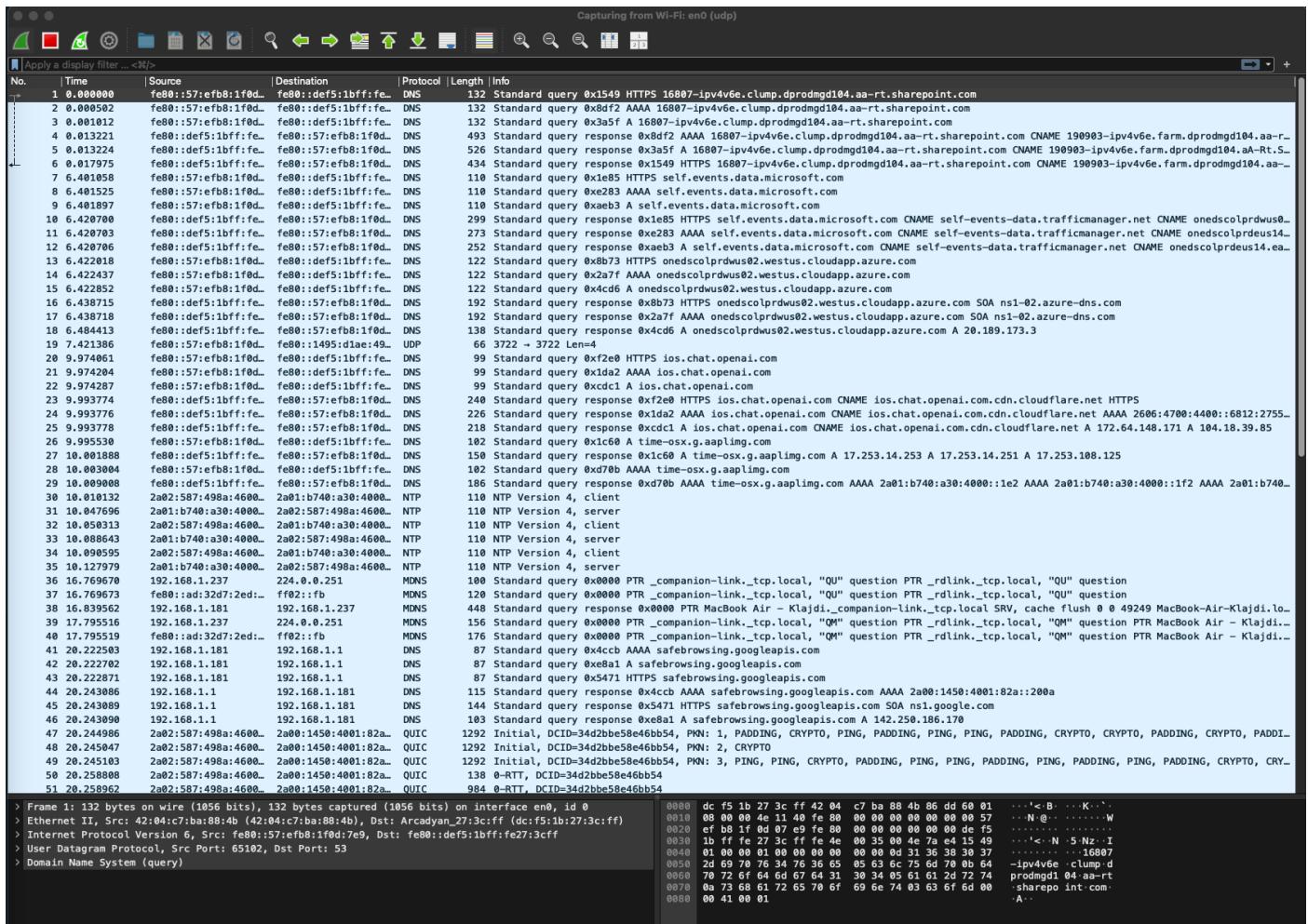
Activity 1: Perform a packet capture using the capture filter a) **tcp** and b) **udp**.

Solution:

a) tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2a02:587:498a:4600..	2a00:1450:400c:c02..	TCP	74	49248 - 5228 [ACK] Seq=1 Ack=1 Win=2048 Len=0
2	0.050236	2a00:1450:400c:c02..	2a02:587:498a:4600..	TCP	86	[TCP ACKed unseen segment] 5228 -> 49248 [ACK] Seq=1 Ack=2 Win=0 TSval=2144224475 TSecr=1463356041
3	6.628289	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	98	54569 - 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=64 TSval=3451142496 TSecr=0 SACK_PERM
4	6.652194	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	86	443 - 54569 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 WS=256 SACK_PERM
5	6.652852	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
6	6.652859	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	591	Client Hello (SNI=aristotleuniversity-my.sharepoint.com)
7	6.670968	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=1 Ack=518 Win=4193792 Len=0
8	6.670972	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	173	HelloRetryRequest, ChangeCipherSpec
9	6.675115	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=518 Ack=100 Win=262016 Len=0
10	6.675122	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	518	ChangeCipherSpec, ClientHello (SNI=aristotleuniversity-my.sharepoint.com)
11	6.692812	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=100 Ack=962 Win=4193280 Len=0
12	6.745779	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	1506	ServerHello
13	6.745783	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	1506	443 - 54569 [ACK] Seq=1532 Ack=962 Win=4193280 Len=1432 [TCP PDU reassembled in 16]
14	6.745786	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	1506	443 - 54569 [ACK] Seq=964 Ack=962 Win=4193280 Len=1432 [TCP PDU reassembled in 16]
15	6.745789	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	1506	443 - 54569 [ACK] Seq=4396 Ack=962 Win=4193280 Len=1432 [TCP PDU reassembled in 16]
16	6.745791	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	574	ApplicationData
17	6.751284	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=962 Ack=6328 Win=262144 Len=0
18	6.759576	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	148	ApplicationData
19	6.761252	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	661	ApplicationData
20	6.761576	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	327	ApplicationData
21	6.761670	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	165	ApplicationData
22	6.775675	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=6328 Ack=1036 Win=4193280 Len=0
23	6.775677	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	533	ApplicationData
24	6.775680	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	136	ApplicationData
25	6.775931	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=1907 Ack=6849 Win=261568 Len=0
26	6.776668	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	185	ApplicationData
27	6.777745	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=6849 Ack=1623 Win=4194304 Len=0
28	6.777748	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	165	ApplicationData
29	6.777750	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=6880 Ack=1907 Win=4193792 Len=0
30	6.777987	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=1938 Ack=6880 Win=262088 Len=0
31	6.792698	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=6880 Ack=1938 Win=4193792 Len=0
32	6.838661	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	1506	443 - 54569 [ACK] Seq=6880 Ack=1938 Win=4193792 Len=1432 [TCP PDU reassembled in 33]
33	6.838664	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	291	ApplicationData
34	6.838666	2620:1ec:8fa::10	2a02:587:498a:4600..	TLSv1..	185	ApplicationData
35	6.838835	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=1938 Ack=8560 Win=260416 Len=0
36	6.847919	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	113	ApplicationData
37	6.848259	2a02:587:498a:4600..	2620:1ec:8fa::10	TLSv1..	98	ApplicationData
38	6.848536	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [FIN, ACK] Seq=2001 Ack=8560 Win=262144 Len=0
39	6.863407	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=8560 Ack=1977 Win=4193792 Len=0
40	6.863410	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=8560 Ack=2001 Win=4193792 Len=0
41	6.863412	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [ACK] Seq=8560 Ack=2002 Win=4193792 Len=0
42	6.865950	2620:1ec:8fa::10	2a02:587:498a:4600..	TCP	74	443 - 54569 [FIN, ACK] Seq=8560 Ack=2002 Win=4193792 Len=0
43	6.866059	2a02:587:498a:4600..	2620:1ec:8fa::10	TCP	74	54569 - 443 [ACK] Seq=2002 Ack=8561 Win=262144 Len=0
44	9.640682	52.123.136.220	192.168.1.181	TCP	54	443 - 50079 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
45	9.939617	192.168.1.181	52.112.238.164	TCP	78	50088 - 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2724329768 TSecr=0 SACK_PERM
46	10.005032	52.112.238.164	192.168.1.181	TCP	74	443 - 50088 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1390 WS=256 SACK_PERM TSval=621570581 TSecr=2724329768
47	10.085677	192.168.1.181	52.112.238.164	TCP	66	50088 - 443 [ACK] Seq=1 Ack=1 Win=132288 Len=0 TSval=2724329834 TSecr=621570581
48	10.085681	192.168.1.181	52.112.238.164	TLSv1..	583	ClientHello (SNI=presence.teams.microsoft.com)
49	10.071672	52.112.238.164	192.168.1.181	TLSv1..	165	HelloRetryRequest, ChangeCipherSpec
50	10.074188	192.168.1.181	52.112.238.164	TCP	66	50088 - 443 [ACK] Seq=518 Ack=100 Win=132160 Len=0 TSval=2724329980 TSecr=621570648
51	10.074113	192.168.1.181	52.112.238.164	TLSv1..	465	ChangeCipherSpec, ClientHello (SNI=presence.teams.microsoft.com)

b) udp



Activity 2: Load the file **capture_data.pcapng**, which can be found on **eLearning**, and apply the following rules using the **display filter**:

- Apply a display filter so that only **IPv4** packets are displayed.
- Apply a display filter so that only **IPv6** packets are displayed.
- Apply a display filter so that only packets using the **ARP protocol** are displayed.
- Apply a display filter so that only packets using the **ARP protocol** and having a destination MAC address of **20:23:51:42:60:8a** are displayed.
- Apply a display filter so that only packets with a **source address of 192.168.1.4** and using the **TCP protocol** are displayed.
- Apply a display filter so that only packets with a **frame size greater than 1458B** are displayed.

Solution:

a)

capture_data.pcapng

ip

No.	Time	Source	Destination	Protocol	Length	Info
3	0.423115	192.168.1.4	213.133.127.245	TCP	66	64686 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4	0.479046	213.133.127.245	192.168.1.4	TCP	66	443 → 64686 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1404 SACK_PERM WS=128
5	0.479151	192.168.1.4	213.133.127.245	TCP	54	64686 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0
6	0.481118	192.168.1.4	213.133.127.245	TLSv1...	571	Client Hello (SNI=matchcenter.to10.gr)
11	0.541737	213.133.127.245	192.168.1.4	TCP	60	443 → 64686 [ACK] Seq=1 Ack=518 Win=30336 Len=0
12	0.544134	213.133.127.245	192.168.1.4	TLSv1...	1458	Server Hello
13	0.545598	213.133.127.245	192.168.1.4	TLSv1...	1281	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
14	0.545673	192.168.1.4	213.133.127.245	TCP	54	64686 → 443 [ACK] Seq=518 Ack=2632 Win=131840 Len=0
15	0.549255	192.168.1.4	213.133.127.245	TLSv1...	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
16	0.607636	213.133.127.245	192.168.1.4	TLSv1...	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
17	0.607636	213.133.127.245	192.168.1.4	TLSv1...	123	Application Data
18	0.607794	192.168.1.4	213.133.127.245	TCP	54	64686 → 443 [ACK] Seq=644 Ack=2959 Win=131584 Len=0
19	0.608149	192.168.1.4	213.133.127.245	TLSv1...	153	Application Data
20	0.608167	192.168.1.4	213.133.127.245	TLSv1...	466	Application Data
21	0.608178	192.168.1.4	213.133.127.245	TLSv1...	125	Application Data
22	0.608186	192.168.1.4	213.133.127.245	TLSv1...	131	Application Data
23	0.608200	192.168.1.4	213.133.127.245	TLSv1...	131	Application Data
24	0.608212	192.168.1.4	213.133.127.245	TLSv1...	131	Application Data
25	0.608350	192.168.1.4	213.133.127.245	TLSv1...	92	Application Data
26	0.665490	213.133.127.245	192.168.1.4	TLSv1...	92	Application Data
27	0.673717	213.133.127.245	192.168.1.4	TCP	1458	443 → 64686 [ACK] Seq=2997 Ack=1155 Win=31360 Len=1404 [TCP PDU reassembled in
28	0.673783	192.168.1.4	213.133.127.245	TCP	54	64686 → 443 [ACK] Seq=1495 Ack=4401 Win=131840 Len=0
29	0.675179	213.133.127.245	192.168.1.4	TCP	1458	443 → 64686 [ACK] Seq=4401 Ack=1155 Win=31360 Len=1404 [TCP PDU reassembled in
30	0.675222	192.168.1.4	213.133.127.245	TCP	54	64686 → 443 [ACK] Seq=1495 Ack=5805 Win=131840 Len=0
31	0.676927	213.133.127.245	192.168.1.4	TCP	1458	443 → 64686 [ACK] Seq=5805 Ack=1155 Win=31360 Len=1404 [TCP PDU reassembled in

```
> Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: zte_4d:38:7b
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 213.133.127.245
> Transmission Control Protocol, Src Port: 64686, Dst Port: 443, Seq: 0, Len: 0
0000 e0 19 54 4d 38 7b a0 d3 c1 32 fd 3c 08 00 45 00 ..TM8{... 2 <`E:
0010 00 34 11 93 40 00 80 06 00 00 c0 a8 01 04 d5 85 .4 @...
0020 7f f5 fc ae 01 bb cd 12 df 8f 00 00 00 00 80 02 .....
0030 fa f0 17 4e 00 00 02 04 05 b4 01 03 03 08 01 01 ...N.....
0040 04 02 ..
```

b)

capture_data.pcapng

ipv6

No.	Time	Source	Destination	Protocol	Length	Info
103	0.920103	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	115	Standard query 0xda18 PTR 1.1.168.192.in-addr.arpa OPT
104	0.920139	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	115	Standard query 0xefa9 PTR 4.1.168.192.in-addr.arpa OPT
105	0.920243	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	119	Standard query 0x0ebc PTR 245.127.133.213.in-addr.arpa OPT
106	0.920380	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	115	Standard query 0x81ff PTR 3.1.168.192.in-addr.arpa OPT
108	0.946451	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	174	Standard query response 0xda18 No such name PTR 1.1.168.192.in-addr.arpa SOA
109	0.947917	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	174	Standard query response 0xefa9 No such name PTR 4.1.168.192.in-addr.arpa SOA
110	0.948906	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	157	Standard query response 0x0ebc PTR 245.127.133.213.in-addr.arpa PTR kronos.al...
111	0.950674	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	174	Standard query response 0x81ff No such name PTR 3.1.168.192.in-addr.arpa SOA
116	1.920153	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	163	Standard query 0x42a3 PTR 6.0.f.f.a.4.2.a.7.3.0.4.2.6.1.7.0.0.1.2.1.1.4.8.c.4
117	1.920261	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	163	Standard query 0xc0ab PTR 3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.5.0.8.4.8.0.0.8.4
118	1.993323	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	195	Standard query response 0xc0ab PTR 3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.5.0.8.4.
119	2.008520	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	231	Standard query response 0x42a3 No such name PTR 6.0.f.f.a.4.2.a.7.3.0.4.2.6.1...
122	2.867444	2a02:214c:8411:210...	2a00:1450:400c:c0b...	TCP	74	64685 → 443 [FIN, ACK] Seq=1 Ack=1 Win=510 Len=0
126	2.932650	2a02:214c:8411:210...	2a02:2148:84:8054:...	DNS	163	Standard query 0xc3f4 PTR 4.5.0.0.0.0.0.0.0.0.0.0.0.0.b.0.c.0.c.0.0.4.0.5
127	2.935467	2a00:1450:400c:c0b...	2a02:214c:8411:210...	TCP	74	443 → 64685 [FIN, ACK] Seq=1 Ack=2 Win=1046 Len=0
128	2.935636	2a02:214c:8411:210...	2a00:1450:400c:c0b...	TCP	74	64685 → 443 [ACK] Seq=2 Ack=2 Win=510 Len=0
129	2.957747	2a02:2148:84:8054:...	2a02:214c:8411:210...	DNS	220	Standard query response 0xc3f4 PTR 4.5.0.b.c.c.0.c...
131	3.997184	2a00:1450:4017:805...	2a02:214c:8411:210...	UDP	141	443 → 64179 Len=79
132	4.006329	2a02:214c:8411:210...	2a00:1450:4017:805...	UDP	95	64179 → 443 Len=33
134	4.198699	2a02:214c:8411:210...	2a00:1450:4017:805...	UDP	91	64179 → 443 Len=29
135	4.240691	2a00:1450:4017:805...	2a02:214c:8411:210...	UDP	87	443 → 64179 Len=25
138	4.442786	2a02:214c:8411:210...	2a00:1450:4017:805...	UDP	91	64179 → 443 Len=29
139	4.486063	2a00:1450:4017:805...	2a02:214c:8411:210...	UDP	87	443 → 64179 Len=25
141	4.686894	2a02:214c:8411:210...	2a00:1450:4017:805...	UDP	91	64179 → 443 Len=29
142	4.729175	2a00:1450:4017:805...	2a02:214c:8411:210...	UDP	87	443 → 64179 Len=25

```
> Frame 103: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: zte_4d:38:7b
> Internet Protocol Version 6, Src: 2a02:214c:8411:2100:7162:4037:a24a:ff06, Dst:
> User Datagram Protocol, Src Port: 55151, Dst Port: 53
> Domain Name System (query)
0000 e0 19 54 4d 38 7b a0 d3 c1 32 fd 3c 86 dd 60 00 ..TM8{... 2 <`...
0010 00 00 00 3d 11 ff 2a 02 21 4c 84 11 21 00 71 62 ..-=.*!L..!qb
0020 40 37 a2 4a ff 06 2a 02 21 48 00 84 80 54 00 00 @7.J.*!H..T...
0030 00 00 00 00 53 d7 6f 00 35 00 3d 10 0f da 18 .....S.o 5=...
0040 01 00 00 01 00 00 00 00 00 01 01 31 01 31 03 31 .....1.1.1...
0050 36 38 03 31 39 32 07 69 6e 2d 61 64 64 72 04 61 68.192.i n-addr.a...
0060 72 70 61 00 00 0c 00 01 00 00 29 04 d0 00 00 00 rpa.....
0070 00 00 00
```

c)

Screenshot of Wireshark showing ARP traffic. The packet list shows several ARP requests and responses. A specific ARP reply from the source 'HewlettPacka_32:fd:3c' to the destination 'TpLinkPte_42:60:8a' at time 0.519796 is highlighted.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	zte_4d:38:7b	HewlettPacka_32:fd...	ARP	60	Who has 192.168.1.4? Tell 192.168.1.1
2	0.000032	HewlettPacka_32:fd...	zte_4d:38:7b	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c
7	0.519758	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.4? Tell 192.168.1.3
8	0.519758	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.7? Tell 192.168.1.3
9	0.519758	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.13? Tell 192.168.1.3
10	0.519796	HewlettPacka_32:fd...	TpLinkPte_42:60:8a	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c
115	1.519789	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.13? Tell 192.168.1.3
120	2.519781	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
121	2.519781	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.13? Tell 192.168.1.3
130	3.519715	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
140	4.519724	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
177	8.519690	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.5? Tell 192.168.1.3
178	9.519690	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.14? Tell 192.168.1.3
179	9.519690	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.5? Tell 192.168.1.3
180	10.519804	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.14? Tell 192.168.1.3
181	10.519804	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.5? Tell 192.168.1.3
184	11.521032	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.14? Tell 192.168.1.3
189	12.519695	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.4? Tell 192.168.1.3
190	12.519728	HewlettPacka_32:fd...	TpLinkPte_42:60:8a	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c
191	16.521939	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
192	16.521939	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.1? Tell 192.168.1.3
193	17.519718	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
221	18.519727	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.8? Tell 192.168.1.3
222	18.519727	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.7? Tell 192.168.1.3
223	19.519717	TpLinkPte_42:60:8a	Broadcast	ARP	60	Who has 192.168.1.6? Tell 192.168.1.3

Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
 Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: TpLinkPte_42
 Address Resolution Protocol (reply)

0000 20 23 51 42 60 8a a0 d3 c1 32 fd 3c 08 06 00 01 #QB`... 2-<...
 0010 08 00 06 04 00 02 a0 d3 c1 32 fd 3c c0 a8 01 04 2-<...
 0020 20 23 51 42 60 8a c0 a8 01 03 #QB`... .

d)

Screenshot of Wireshark showing ARP traffic. The packet list shows three ARP frames. The first frame is the ARP reply from the source 'HewlettPacka_32:fd:3c' to the destination 'TpLinkPte_42:60:8a' at time 0.519796. The second frame is the ARP request from the source 'HewlettPacka_32:fd:3c' to the destination 'TpLinkPte_42:60:8a' at time 12.519728. The third frame is another ARP request from the source 'HewlettPacka_32:fd:3c' to the destination 'TpLinkPte_42:60:8a' at time 24.519747.

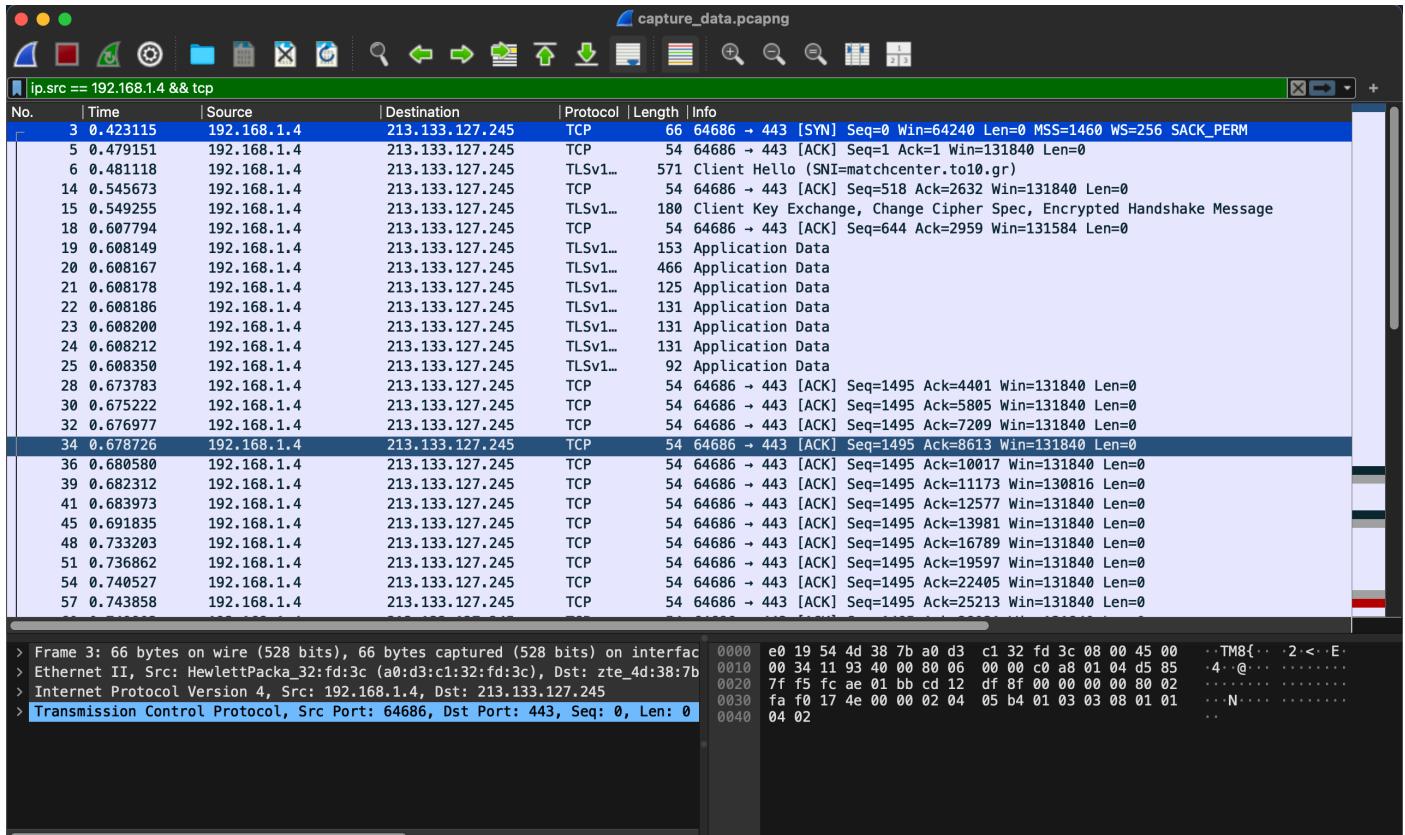
No.	Time	Source	Destination	Protocol	Length	Info
10	0.519796	HewlettPacka_32:fd...	TpLinkPte_42:60:8a	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c
190	12.519728	HewlettPacka_32:fd...	TpLinkPte_42:60:8a	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c
333	24.519747	HewlettPacka_32:fd...	TpLinkPte_42:60:8a	ARP	42	192.168.1.4 is at a0:d3:c1:32:fd:3c

Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
 Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: TpLinkPte_42:60:8a
 Address Resolution Protocol (reply)

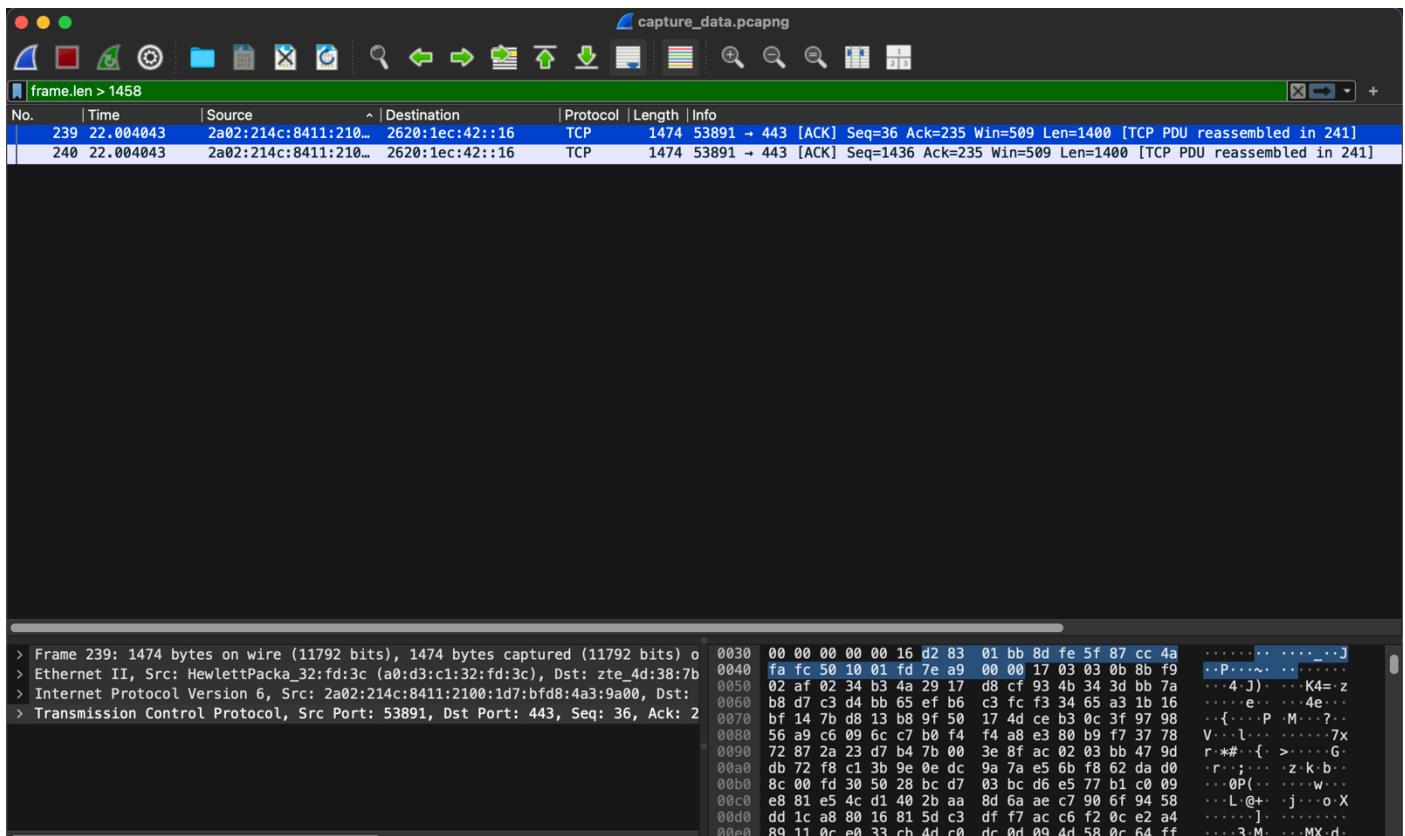
Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: reply (2)
 Sender MAC address: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c)
 Sender IP address: 192.168.1.4

0000 20 23 51 42 60 8a a0 d3 c1 32 fd 3c 08 06 00 01 #QB`... 2-<...
 0010 08 00 06 04 00 02 a0 d3 c1 32 fd 3c c0 a8 01 04 2-<...
 0020 20 23 51 42 60 8a c0 a8 01 03 #QB`... .

e)



f)

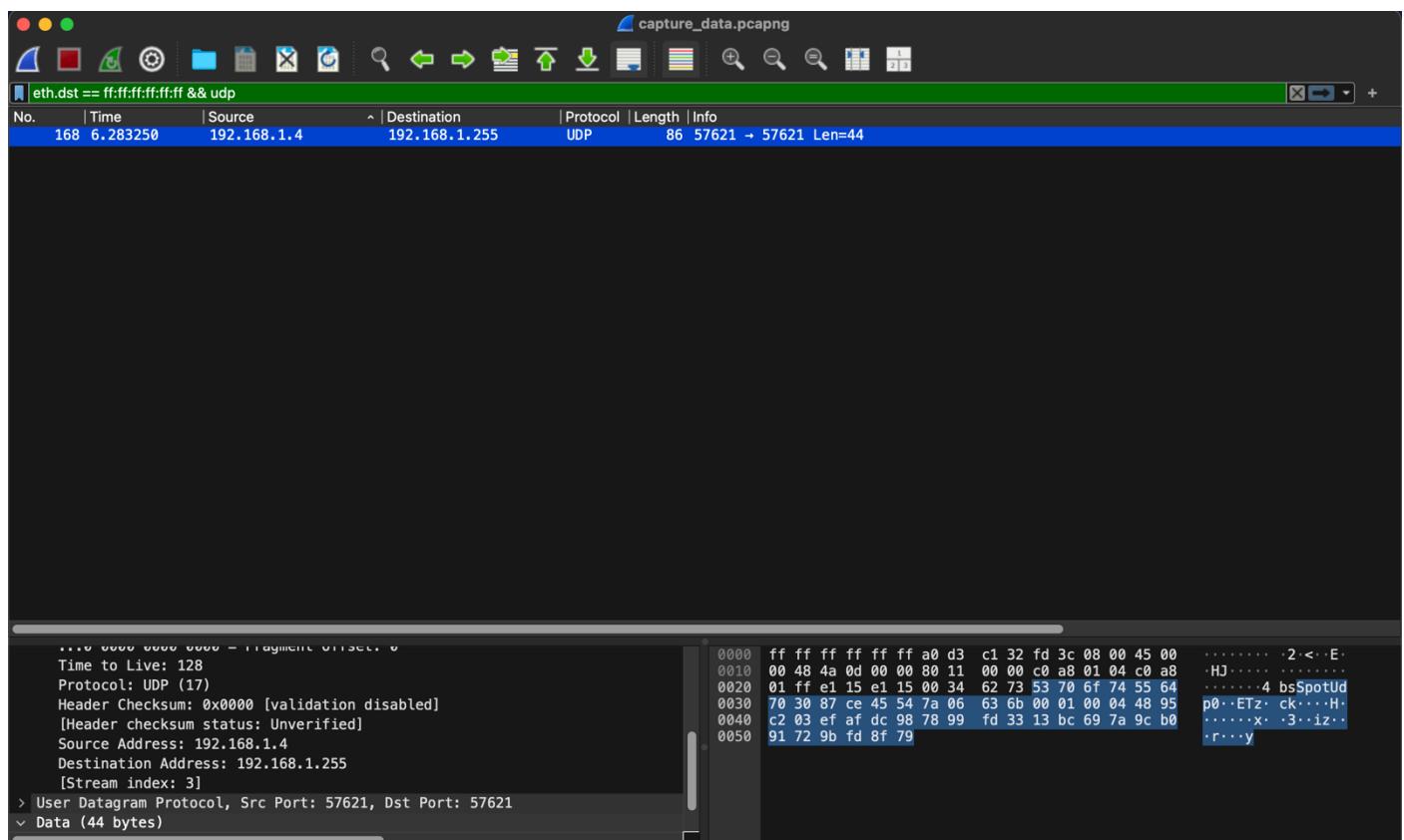


Activity 3: In the same file (**capture_data.pcapng**):

- a) Apply a display filter so that only packets with a **destination MAC address** set to the **broadcast address** and using the **UDP protocol** are displayed.
- b) By exploring the **single packet** that results from the filter using the **packet details pane**, report the following:
- i. **Header length** (IP layer)
 - ii. **Total packet size** (IP layer)
 - iii. **Source MAC address** (Ethernet layer)
 - iv. **Source and destination IP addresses**
 - v. **Source and destination port numbers** (Transport layer - UDP)
 - vi. **Header length** (Transport layer - UDP)
 - vii. **Total packet size** (Transport layer - UDP)
 - viii. **Payload size** (Transport layer - UDP)

Solution:

a)



b)

i) 20 Bytes

```
> Frame 168: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{432A4B18-D242-49ED-AC90-6ADE56F2872F}, id 0
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
< Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 72
        Identification: 0x4a0d (18957)
    > 000. .... = Flags: 0x0
        ... 0000 0000 0000 = Fragment Offset: 0
```

ii) 72

iii) ff:ff:ff:ff:ff:ff

```
> Frame 168: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{432A4B18-D242-49ED-AC90-6ADE56F2872F}, id 0
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
< Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    ....1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
    ....1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    > Source: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c)
        Type: IPv4 (0x0800)
        [Stream index: 3]
    > Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.255
    > User Datagram Protocol, Src Port: 57621, Dst Port: 57621
```

iv) 192:168:1.4

```
> Frame 168: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{432A4B18-D242-49ED-AC90-6ADE56F2872F}, id 0
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
< Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 72
        Identification: 0x4a0d (18957)
    > 000. .... = Flags: 0x0
        ...0 0000 0000 0000 = Fragment Offset: 0
```

v) 57621

```
> Frame 168: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{432A4B18-D242-49ED-AC90-6ADE56F2872F}, id 0
> Ethernet II, Src: HewlettPacka_32:fd:3c (a0:d3:c1:32:fd:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.255
< User Datagram Protocol, Src Port: 57621, Dst Port: 57621
    Source Port: 57621
    Destination Port: 57621
    Length: 52
    Checksum: 0x6273 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
```

vi) 8

vii) 44

viii) 52

Activity 4: Load the file **http_capture.pcapng**, which can be found on **eLearning**, and apply the following rules using the **display filter**:

a) Apply a display filter so that only packets using the **HTTP protocol** and having a **packet size of 301** are displayed.

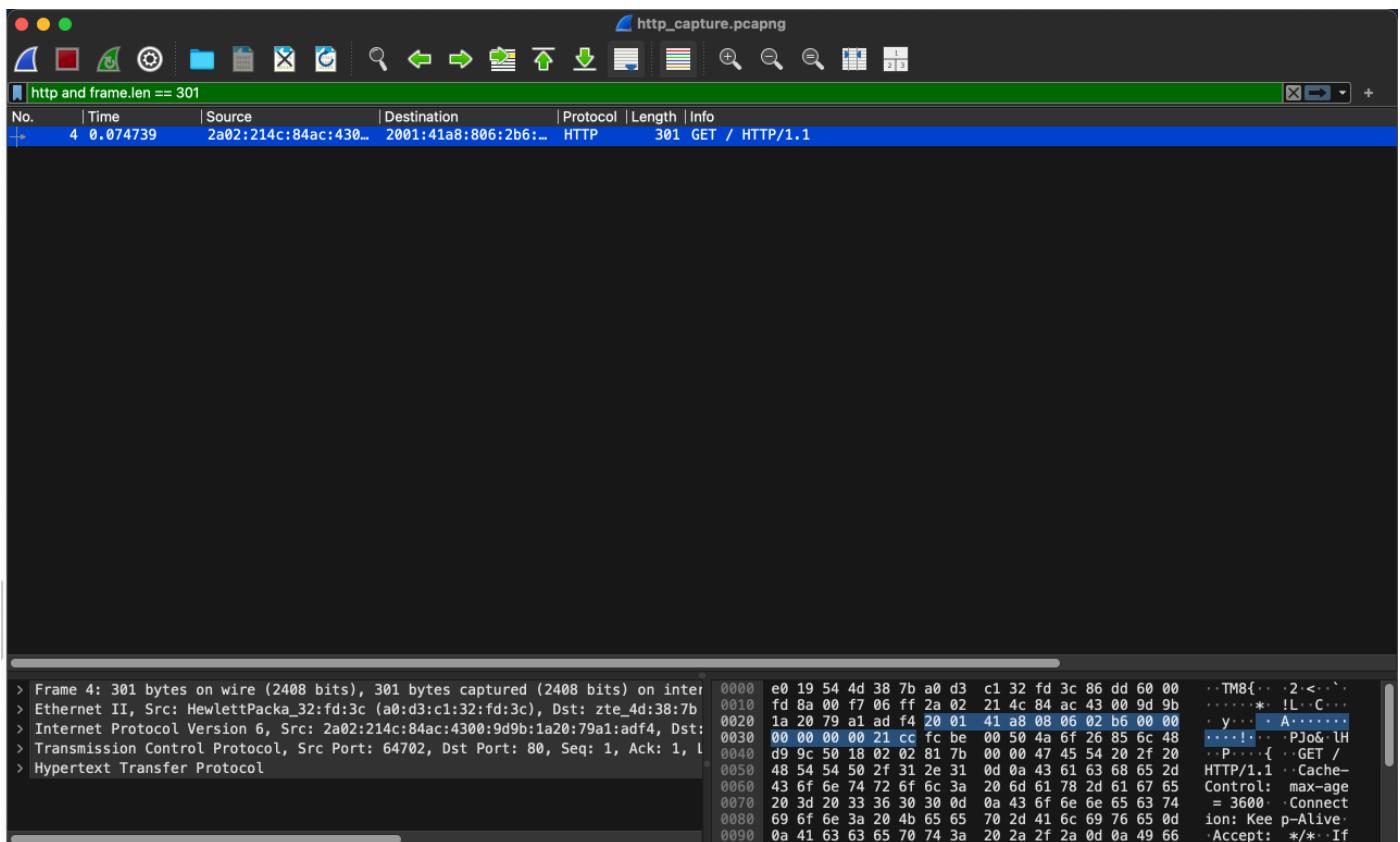
b) By exploring the **single packet** that results from the filter using the **packet and byte details pane**, report the following:

i. Provide in **hexadecimal format** the values of **Next Header** and **Hop Limit** (IP layer).

ii. Provide in **hexadecimal format** the **source and destination port numbers**, as well as the **sequence number** for this packet (Transport Layer).

Solution:

a)



b)

i)

Next Header: 0x06 (TCP)

Hop Limit: 0xFF (255)

ii)

Field	Decimal Format	Hexadecimal Format
Source Port	64702	0xFCEE
Destination Port	80	0x0050

Sequence Number

1248798341

0x4A7F7C65

Nmap

Activity 5: Perform a **port scanning** on the domain **www.in.gr** using the methods mentioned in the previous slides. Which of the methods was the most time-consuming and why?

Solution:

1) SYN Scan (`sudo nmap -sS www.in.gr`)

Advantages:

Fast, because it does not complete the **TCP handshake**.

Stealthy, meaning it is hard to detect by the server.

Disadvantages:

Requires **root privileges** (`sudo`).

Can be **filtered** by **firewalls** and **IDS**.

2) TCP Connect Scan (`nmap -sT www.in.gr`)

Advantages:

Does **not require root privileges**.

More **reliable**, as it completes the **TCP handshake**.

Disadvantages:

Slower than SYN Scan, as it performs a full **TCP handshake (SYN-ACK-ACK)**.

Easier to detect by **firewalls** and **IDS**.

3) ACK Scan (`sudo nmap -sA www.in.gr`)

Advantages:

Useful for **firewall detection** (determines if a port is **filtered** or not).

Does **not scan for open/closed ports**, but checks if a firewall exists.

Disadvantages:

Cannot detect if a port is **open or closed**—only if it is **filtered**.

Does not reveal active services, as it does not receive standard responses.

4) UDP Scan (sudo nmap -sU www.in.gr)

Advantages:

Useful for detecting **UDP services** (e.g., **DNS, DHCP, SNMP, NTP**).

Used when looking for **services that are not detected** in TCP scanning.

Disadvantages:

The slowest method, because **UDP lacks a SYN-ACK mechanism**, meaning **Nmap must wait for timeouts**.

If a **firewall** is present, **UDP packets may be filtered**, making it even **slower**.

Cannot always detect closed ports, due to a lack of responses.