

Spring 2025

Cyber Security Fundamentals (INFT-3508 - 20332)

Aysel Panahova

Quiz 6

Task 1:

Firstly, I opened the WireShark and it started capturing traffic. As written in the instructions, we need to analyze HTTP requests, so I closed all my tabs and visited a single website that uses HTTP protocol (rather than HTTPS, which is encrypted and does not reveal readable data) in my browser. The website was “<http://httpforever.com>” This ensured that WireShark could capture HTTP requests for analysis.

I entered “HTTP” in the filter bar to display only HTTP traffic. This helped eliminate unrelated network protocols from the view, making it easier to focus on the HTTP request specifically.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------|---------------|---------------|----------|--------|-----------------------------|
| 354 | 13:53:58.545502 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 357 | 13:53:58.947210 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 369 | 13:54:03.778835 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 373 | 13:54:04.012000 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |

Task 2:

From the screenshot, we can see that there are two pairs of requests and responses. We can calculate the time between the GET requests and the responses of these pairs.

First pair:

Frame 354 - GET request: 58.545502 sec

Frame 357 – HTTP OK: 58.947210 sec

Time difference: 0.401708 sec

Second Pair:

Frame 369 - GET request: 03.778835 sec

Frame 373 – HTTP OK: 04.012000 sec

Time difference: 0.233165 sec

The time between HTTP GET and response varies between approximately 0.40, 0.23 seconds.

Task: 3

Upon analyzing the packet capture, I identified 28 HTTP GET requests directed to a remote website, using the filter ip.dst == 146.190.62.39 && http. Additionally, the traffic reveals three distinct destination IP addresses: 34.223.124.45, 146.190.62.39, and 2.20.77.60, indicating that multiple external servers were contacted during the browsing session.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-----------------|---------------|---------------|----------|--------|---|
| 354 | 13:53:56.945302 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 357 | 13:53:58.947210 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 369 | 13:54:03.778835 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 373 | 13:54:04.812080 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 434 | 13:54:15.101477 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 439 | 13:54:15.533804 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 6611 | 14:01:10.254234 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 6615 | 14:01:10.662760 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 9304 | 14:03:37.577599 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9311 | 14:03:37.805880 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 9361 | 14:03:48.478182 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9365 | 14:03:48.753208 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 9426 | 14:03:55.290310 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9431 | 14:03:55.598200 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 21377 | 14:05:09.838339 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 21383 | 14:05:09.264894 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 23844 | 14:06:15.031719 | 10.0.110.91 | 104.81.99.218 | HTTP | 425 | GET /ME8wTTBUMEkwRzAHBgUrDgMCggQU646nUncrcfKRxkj8qXxwcUeV7UEFLPbSKT5ocXYrjZBzBFjaWIpvEvGAhAMc |
| 28450 | 14:07:33.806377 | 10.0.110.91 | 34.223.124.45 | HTTP | 439 | GET / HTTP/1.1 |
| 28453 | 14:07:34.126386 | 34.223.124.45 | 10.0.110.91 | HTTP | 973 | HTTP/1.1 200 OK (text/html) |
| 28462 | 14:07:34.739695 | 10.0.110.91 | 34.223.124.45 | HTTP | 498 | GET /online HTTP/1.1 |
| 28464 | 14:07:35.148170 | 34.223.124.45 | 10.0.110.91 | HTTP | 603 | HTTP/1.1 301 Moved Permanently (text/html) |
| 28466 | 14:07:35.160829 | 10.0.110.91 | 34.223.124.45 | HTTP | 499 | GET /online/ HTTP/1.1 |
| 28468 | 14:07:35.559497 | 34.223.124.45 | 10.0.110.91 | HTTP | 217 | HTTP/1.1 200 OK (text/html) |
| 28474 | 14:07:35.704762 | 10.0.110.91 | 34.223.124.45 | HTTP | 442 | GET /favicon.ico HTTP/1.1 |
| 28475 | 14:07:35.965944 | 34.223.124.45 | 10.0.110.91 | HTTP | 482 | HTTP/1.1 200 OK (PNG) |
| 28875 | 14:08:05.869669 | 10.0.110.91 | 34.223.124.45 | HTTP | 499 | GET /online/ HTTP/1.1 |
| 28887 | 14:08:06.104315 | 34.223.124.45 | 10.0.110.91 | HTTP | 324 | HTTP/1.1 200 OK (text/html) |
| 42504 | 14:23:01.118464 | 10.0.110.91 | 34.223.124.45 | HTTP | 499 | GET /online/ HTTP/1.1 |
| 42507 | 14:23:01.404017 | 34.223.124.45 | 10.0.110.91 | HTTP | 324 | HTTP/1.1 200 OK (text/html) |
| 42522 | 14:23:04.346778 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 42526 | 14:23:04.575060 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 42754 | 14:23:31.848682 | 10.0.110.91 | 146.190.62.39 | HTTP | 425 | GET /css/style-mobile.min.css HTTP/1.1 |
| 42764 | 14:23:32.116939 | 146.190.62.39 | 10.0.110.91 | HTTP | 99 | HTTP/1.1 200 OK (text/css) |
| 78772 | 14:33:15.474490 | 10.0.110.91 | 146.190.62.39 | HTTP | 557 | GET / HTTP/1.1 |
| 78795 | 14:33:15.713434 | 146.190.62.39 | 10.0.110.91 | HTTP | 70 | HTTP/1.1 200 OK (text/html) |
| 78805 | 14:33:15.737251 | 10.0.110.91 | 146.190.62.39 | HTTP | 409 | GET /js/init.min.js HTTP/1.1 |
| 78878 | 14:33:15.966936 | 146.190.62.39 | 10.0.110.91 | HTTP | 528 | HTTP/1.1 200 OK (application/javascript) |
| 78890 | 14:33:16.030901 | 10.0.110.91 | 146.190.62.39 | HTTP | 427 | GET /css/style.min.css HTTP/1.1 |
| 78891 | 14:33:16.030982 | 10.0.110.91 | 146.190.62.39 | HTTP | 432 | GET /css/style-wide.min.css HTTP/1.1 |
| 78902 | 14:33:16.193961 | 10.0.110.91 | 2.20.77.60 | HTTP | 428 | GET / HTTP/1.1 |
| 78904 | 14:33:16.249553 | 2.20.77.60 | 10.0.110.91 | HTTP | 331 | HTTP/1.1 304 Not Modified |
| 78907 | 14:33:16.253754 | 146.190.62.39 | 10.0.110.91 | HTTP | 968 | HTTP/1.1 200 OK (text/css) |
| 78928 | 14:33:16.316313 | 10.0.110.91 | 2.20.77.60 | HTTP | 428 | GET / HTTP/1.1 |

> Frame 324: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits) on interface en0 0:000 176 51 166 246 182 160 156 62 83 113 22 39 8 0 69 0 -3...> Sq...E...
 ● Hypertext Transfer Protocol: Protocol Packets: 254338 - Displayed: 56 (0.0%) · Marked: 3 (0.0%) · Dropped: 0 (0.0%) Profile: Classic

Task: 4

As shown in the picture, there are 4 HTTP GET requests that are images (frame 78996, 79000, 79154, 79160).

| | | | | | | | |
|-------|-----------------|---------------|---------------|----------|------|---|----------------|
| 78996 | 14:33:16.811274 | 10.0.110.91 | 146.190.62.39 | HTTP | 494 | GET /css/images/banner.svg | HTTP/1.1 |
| 79000 | 14:33:16.813736 | 10.0.110.91 | 146.190.62.39 | HTTP | 509 | GET /css/images/header-major-on-light.svg | HTTP/1.1 |
| 79150 | 14:33:17.039826 | 146.190.62.39 | 10.0.110.91 | HTTP/... | 1377 | HTTP/1.1 200 OK | |
| 79154 | 14:33:17.040967 | 10.0.110.91 | 146.190.62.39 | HTTP | 467 | GET /favicon.ico | HTTP/1.1 |
| 79160 | 14:33:17.043598 | 10.0.110.91 | 146.190.62.39 | HTTP | 508 | GET /css/images/header-major-on-dark.svg | HTTP/1.1 |
| 79161 | 14:33:17.045245 | 146.190.62.39 | 10.0.110.91 | HTTP/... | 1327 | HTTP/1.1 200 OK | |
| 79201 | 14:33:17.271995 | 146.190.62.39 | 10.0.110.91 | HTTP | 1059 | HTTP/1.1 200 OK | (image/x-icon) |
| 79203 | 14:33:17.271997 | 146.190.62.39 | 10.0.110.91 | HTTP/... | 1333 | HTTP/1.1 200 OK | |

The images are downloaded in parallel (simultaneously). We can see this because the packet stamps show quick back-and-forth communication between my computer (10.0.110.91) and the website (146.198.62.39). For example, the server sends a packet, then my pc immediately sends another, and this keeps happening in an overlapping pattern. If the downloads were happening one at the same time (serially), we would see long pauses between requests for each image. Instead, the rapid switching between sending and receiving confirms that multiple images were being downloaded at the same time.

Task: 5

Frames 78996, 79000, 79154, and 79160 show packets sent from the local machine which is my pc (10.0.110.91) containing HTTP GET requests. Frames 79150, 79161, 79201, and 79203 show response packets from the remote server (146.190.62.39). This makes a total of eight packets exchanged, meaning four requests from the client matched by four responses from the server.

Task: 6

- a) The TCP connection was established using the standard three-way handshake, which involves three packets:
1. SYN (synchronize) → Initiated by the client (10.0.110.91) to request a connection.
 2. SYN-ACK (synchronize-acknowledge) → Sent by the server (146.190.62.39) to acknowledge the request and propose its own connection parameters.
 3. ACK (acknowledge) → Final confirmation from the client to establish the connection.

This exchange is clearly observed in frames 153, 183, and 185, confirming the connection setup was successful and adhered to TCP protocol standards.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------|---------------|---------------|----------|--------|---|
| 153 | 13:53:37.449237 | 10.0.110.91 | 146.190.62.39 | TCP | 78 | 63666 - > 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 Len=6 TSval=3909552112 TSecr=0 SACK_PERM |
| 183 | 13:53:37.672598 | 146.190.62.39 | 10.0.110.91 | TCP | 74 | 80 -> 63666 [SYN, ACK] Seq=1 Win=65160 Len=0 MSS=1380 SACK_PERM TSval=1541624041 TSecr=3909552336 |
| 185 | 13:53:37.672833 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63666 - > 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=3909552336 TSecr=1541624041 |

- b) To calculate the number of packets that were sent we can use the formula “tcp.len > 0”. This filter shows only the TCP packets that have a payload, meaning the packets that contain actual data, not just control information like ACK or handshakes.

In this case, the result shows 140,580 packets.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------|----------------|----------------|----------|--------|--|
| 4 | 13:53:38.810854 | 216.58.287.202 | 10.0.110.91 | TLSv1_1 | 348 | Application Data |
| 17 | 13:53:33.188693 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 583 | Cipher Hello (SM2+AES-CBC-Hmac-SHA256) |
| 18 | 13:53:33.188700 | 52.189.28.47 | 10.0.110.91 | TLSv1_1 | 117 | Application Data |
| 19 | 13:53:33.282319 | 10.0.110.91 | 54.212.246.137 | TLSv1_1 | 58 | Application Data |
| 20 | 13:53:33.286474 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 127 | Application Data |
| 21 | 13:53:33.286535 | 10.0.110.91 | 34.280.167.42 | TLSv1_1 | 131 | Application Data |
| 22 | 13:53:33.286542 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 127 | Application Data |
| 23 | 13:53:33.286709 | 10.0.110.91 | 34.280.167.42 | TLSv1_1 | 182 | Application Data |
| 24 | 13:53:33.225986 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 127 | Application Data |
| 25 | 13:53:33.226626 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 619 | Application Data |
| 26 | 13:53:33.226633 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 140 | Application Data |
| 27 | 13:53:33.281809 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 466 | Change Cipher Spec, Client Hello (SM2+AES-CBC-Hmac-SHA256) |
| 28 | 13:53:33.282387 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 127 | Application Data |
| 29 | 13:53:33.284150 | 10.0.110.91 | 34.280.167.42 | TLSv1_1 | 127 | Application Data |
| 30 | 13:53:33.284157 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 127 | Application Data |
| 31 | 13:53:33.284302 | 10.0.110.91 | 44.212.246.137 | TCP | 443 | 63638 - 443 [PSH, ACK] Seq=1653 Ack=1 Win=2048 Len=1368 TSval=678164279 TSecr=3832004546 [TCP POU] |
| 32 | 13:53:33.284302 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 746 | 63638 - 443 [PSH, ACK] Seq=3821 Ack=1 Win=2048 Len=608 TSval=678164279 TSecr=3832004546 [TCP F |
| 33 | 13:53:33.284333 | 10.0.110.91 | 44.212.246.137 | TCP | 531 | 63638 - 443 [PSH, ACK] Seq=1891 Ack=1 Win=2048 Len=1368 TSval=781079126 TSecr=3794188761 [TCP POU] |
| 34 | 13:53:33.284333 | 10.0.110.91 | 34.280.167.42 | TCP | 543 | 63638 - 443 [PSH, ACK] Seq=2459 Ack=1 Win=2048 Len=608 TSval=781079126 TSecr=3794188761 [TCP F |
| 35 | 13:53:33.284521 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 746 | 63638 - 443 [PSH, ACK] Seq=2459 Ack=1 Win=2048 Len=608 TSval=781079126 TSecr=3794188761 [TCP F |
| 36 | 13:53:33.284522 | 10.0.110.91 | 34.280.167.42 | TLSv1_1 | 379 | Application Data |
| 37 | 13:53:33.378637 | 54.210.191.50 | 10.0.110.91 | TLSv1_1 | 98 | Application Data |
| 38 | 13:53:33.378637 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 189 | Application Data |
| 39 | 13:53:33.378637 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 239 | Application Data |
| 40 | 13:53:33.378637 | 54.210.191.50 | 10.0.110.91 | TLSv1_1 | 189 | Application Data |
| 41 | 13:53:33.377272 | 34.280.167.42 | 10.0.110.91 | TLSv1_1 | 189 | Application Data |
| 42 | 13:53:33.377272 | 44.212.246.137 | 10.0.110.91 | TLSv1_1 | 184 | Application Data |
| 43 | 13:53:33.377272 | 34.280.167.42 | 10.0.110.91 | TLSv1_1 | 189 | Application Data |
| 44 | 13:53:33.377272 | 34.280.167.42 | 44.212.246.137 | TLSv1_1 | 184 | Application Data |
| 45 | 13:53:33.377272 | 10.0.110.91 | 44.212.246.137 | TCP | 443 | 63664 - 63664 [ACK] Seq=1468 Ack=918 Win=4194560 Len=1368 TSval=2033817451 TSecr=1339805317 [TCP F |
| 46 | 13:53:33.378646 | 52.189.28.47 | 10.0.110.91 | TCP | 1434 | 443 - 63664 [ACK] Seq=1468 Ack=918 Win=4194560 Len=1368 TSval=2033817451 TSecr=1339805317 [TCP F |
| 47 | 13:53:33.378646 | 52.189.28.47 | 44.212.246.137 | TLSv1_1 | 608 | Application Data |
| 48 | 13:53:33.378647 | 52.189.28.47 | 10.0.110.91 | TCP | 1434 | 443 - 63664 [ACK] Seq=2356 Ack=918 Win=4194560 Len=1368 TSval=2033817451 TSecr=1339805317 [TCP F |
| 49 | 13:53:33.378647 | 52.189.28.47 | 44.212.246.137 | TLSv1_1 | 608 | Application Data |
| 50 | 13:53:33.378647 | 10.0.110.91 | 44.212.246.137 | TLSv1_1 | 608 | Application Data |
| 51 | 13:53:33.385505 | 44.212.246.137 | 10.0.110.91 | TLSv1_1 | 368 | Application Data |
| 52 | 13:53:33.385338 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 149 | Application Data |
| 53 | 13:53:33.385338 | 52.189.28.47 | 10.0.110.91 | TLSv1_1 | 368 | Application Data |
| 54 | 13:53:33.385338 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 368 | Application Data |
| 55 | 13:53:33.385338 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 368 | Application Data |
| 56 | 13:53:33.385338 | 10.0.110.91 | 52.189.28.47 | TLSv1_1 | 368 | Application Data |

> Frame 534: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface en0, 1 link layer frames received
 Ethernet II, Src: Apple_71:61:27 (02:00:00:71:61:27), Dst: Wi-Fi Adapter (00:0c:29:1e:00:00)
 Internet Protocol Version 4, Src: 10.0.110.91, Dst: 146.190.62.39
 Transmission Control Protocol, Src Port: 40981, Seq: 12, Ack: 12, Len: 11

Packets: 254338 - Displayed: 140580 (55.9%) - Marked: 3 (0.0%) - Dropped: 0 (0.0%) - Profile: Classic

c) To calculate the number of acknowledgment packets I used the formula “ip.dst == 146.190.62.39 && tcp.len > 0”.

- ip.dst == 146.190.62.39 means the packet is being sent to that IP address.
- tcp.len > 0 means the packet has a data payload (not just a control or acknowledgment packet).

There are 22 packets.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-----------------|-------------|---------------|----------|--------|---|
| 324 | 13:53:52.128997 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 324 | 13:53:58.545502 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 369 | 13:54:03.778835 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 434 | 13:54:15.181477 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 6611 | 14:01:10.254234 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9384 | 14:03:37.577599 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9361 | 14:03:48.478182 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 9426 | 14:03:55.298310 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 21377 | 14:05:09.093839 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 42522 | 14:23:04.346778 | 10.0.110.91 | 146.190.62.39 | HTTP | 442 | GET / HTTP/1.1 |
| 42754 | 14:23:31.848682 | 10.0.110.91 | 146.190.62.39 | HTTP | 425 | GET /css/style-mobile.min.css HTTP/1.1 |
| 78772 | 14:33:15.474498 | 10.0.110.91 | 146.190.62.39 | HTTP | 557 | GET / HTTP/1.1 |
| 78805 | 14:33:15.737251 | 10.0.110.91 | 146.190.62.39 | HTTP | 409 | GET /js/init.min.js HTTP/1.1 |
| 78809 | 14:33:16.038958 | 10.0.110.91 | 146.190.62.39 | HTTP | 427 | GET /css/style.min.css HTTP/1.1 |
| 78809 | 14:33:16.038958 | 10.0.110.91 | 146.190.62.39 | HTTP | 427 | GET /css/styleguide.min.css HTTP/1.1 |
| 78996 | 14:33:16.011274 | 10.0.110.91 | 146.190.62.39 | HTTP | 494 | GET /css/images/header-major-on-light.svg HTTP/1.1 |
| 79000 | 14:33:16.013736 | 10.0.110.91 | 146.190.62.39 | HTTP | 589 | GET /css/images/header-major-on-light.svg HTTP/1.1 |
| 79154 | 14:33:17.048067 | 10.0.110.91 | 146.190.62.39 | HTTP | 467 | GET /favicon.ico HTTP/1.1 |
| 79160 | 14:33:17.048398 | 10.0.110.91 | 146.190.62.39 | HTTP | 588 | GET /css/images/header-major-on-dark.svg HTTP/1.1 |
| 78580 | 14:33:12.788432 | 10.0.110.91 | 146.190.62.39 | TCP | 1434 | 50166 -> 443 [ACK] Seq=1 Ack=1 Win=131328 Len=1368 TStamp=769884314 TSecr=1543998962 [TCP PDU reasemb |
| 78581 | 14:33:12.788449 | 10.0.110.91 | 146.190.62.39 | TLSv1_ | 519 | Client Hello (SNI=tptfrever.com) |
| 78948 | 14:33:16.394634 | 10.0.110.91 | 146.190.62.39 | TLSv1_ | 96 | Change Cipher Spec, Application Data |

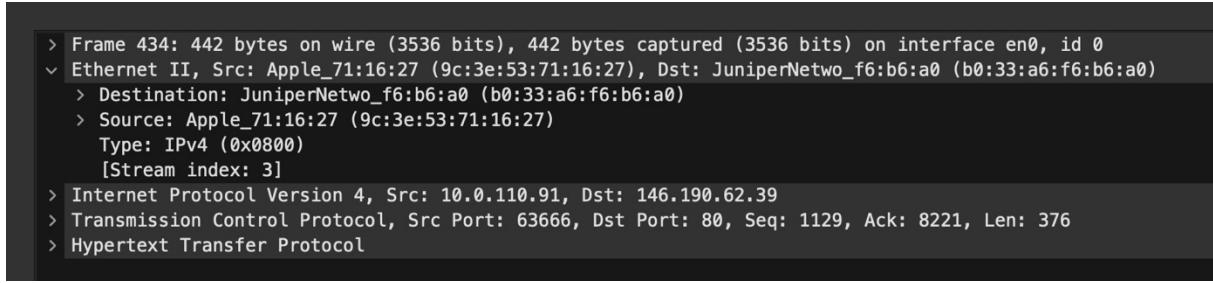
d) The TCP connection closed with these packets:

1. Client (10.0.110.91) sent FIN-ACK to signal it was done sending data.
2. Server (146.190.62.39) replied with FIN-ACK to confirm and close its side.

I used the formula “ip.dst == 146.190.62.39 && tcp.flags.fin == 1”. There appear 12 frames. This filter formula does two things: First, it looks for network packets going to the IP address 146.100.62.39 (the website). Second, it checks if those packets have the TCP FIN flag turned on. The FIN flag means the sender wants to close the connection.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-----------------|-------------|---------------|----------|--------|---|
| 933 | 13:55:10.424539 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63666 -> 80 [FIN, ACK] Seq=1505 Ack=10961 Win=131072 Len=0 TStamp=3909645089 TSecr=1541661703 |
| 7450 | 14:01:40.827356 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63716 -> 80 [FIN, ACK] Seq=377 Ack=2741 Win=131072 Len=0 TStamp=1475169772 TSecr=1542076859 |
| 21115 | 14:04:38.817483 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63729 -> 80 [FIN, ACK] Seq=1129 Ack=8221 Win=131072 Len=0 TStamp=2040449064 TSecr=1542241828 |
| 21911 | 14:05:39.810976 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63773 -> 80 [FIN, ACK] Seq=377 Ack=2741 Win=131072 Len=0 TStamp=32880592532 TSecr=1542315566 |
| 28283 | 14:07:23.373952 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63809 -> 80 [FIN, ACK] Seq=1 Ack=1 Win=131328 Len=0 TStamp=824521345 TSecr=1542419478 |
| 54858 | 14:24:32.889732 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 63910 -> 80 [FIN, ACK] Seq=736 Ack=4142 Win=131072 Len=0 TStamp=1735242460 TSecr=1543418394 |
| 73448 | 14:31:32.575750 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 64000 -> 80 [FIN, ACK] Seq=1 Ack=1 Win=131328 Len=0 TStamp=1819263074 TSecr=1543868120 |
| 78941 | 14:33:16.395330 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 50166 -> 443 [FIN, ACK] Seq=1852 Ack=4221 Win=131072 Len=0 TStamp=769880001 TSecr=1543999193 |
| 78970 | 14:33:16.624017 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | [TCP Retransmission] 50166 -> 443 [FIN, ACK] Seq=1852 Ack=4222 Win=131072 Len=0 TStamp=769888230 TSecr=1543999193 |
| 83571 | 14:35:49.778387 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 50165 -> 80 [FIN, ACK] Seq=810 Ack=2165 Win=131072 Len=0 TStamp=1657260421 TSecr=1544123764 |
| 83572 | 14:35:49.778390 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 50172 -> 80 [FIN, ACK] Seq=443 Ack=1269 Win=131072 Len=0 TStamp=937484349 TSecr=1544123767 |
| 83573 | 14:35:49.778401 | 10.0.110.91 | 146.190.62.39 | TCP | 66 | 50164 -> 80 [FIN, ACK] Seq=2025 Ack=29662 Win=131072 Len=0 TStamp=578498318 TSecr=1544123765 |

Task: 7

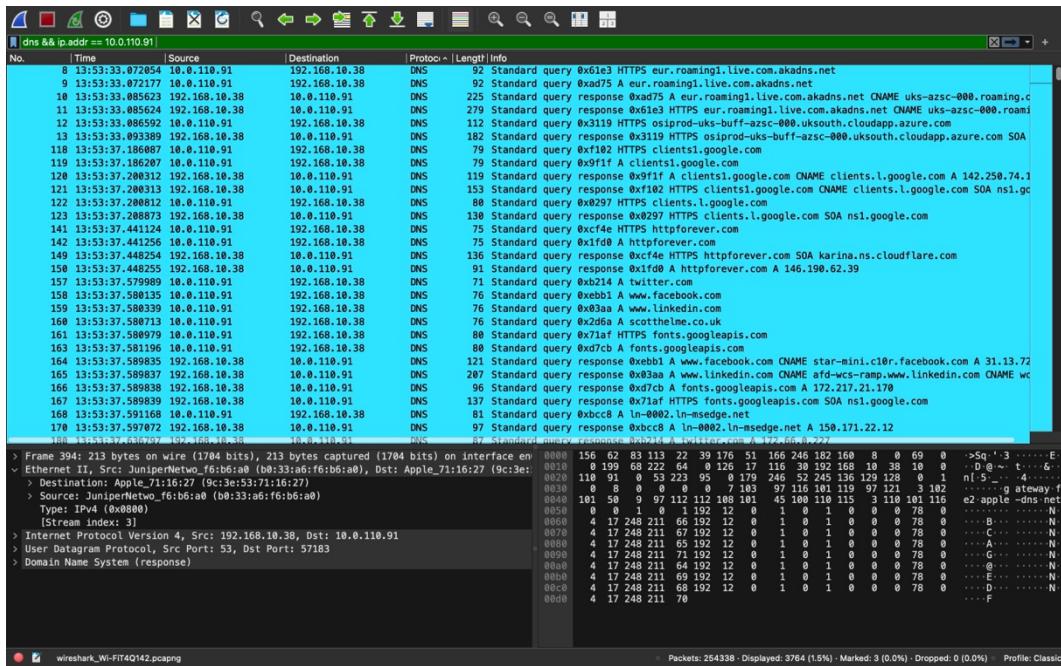


The screenshot shows a Wireshark window with a selected packet. The packet details pane displays the following information:

- Frame 434: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits) on interface en0, id 0
- Ethernet II, Src: Apple_71:16:27 (9c:3e:53:71:16:27), Dst: JuniperNetwo_f6:b6:a0 (b0:33:a6:f6:b6:a0)
- Destination: JuniperNetwo_f6:b6:a0 (b0:33:a6:f6:b6:a0)
- Source: Apple_71:16:27 (9c:3e:53:71:16:27)
- Type: IPv4 (0x0800)
- [Stream index: 3]
- Internet Protocol Version 4, Src: 10.0.110.91, Dst: 146.190.62.39
- Transmission Control Protocol, Src Port: 63666, Dst Port: 80, Seq: 1129, Ack: 8221, Len: 376
- Hypertext Transfer Protocol

I clicked on any packet on the Wireshark and it showed all the details about that packet. As we see from the screenshot, the source, which is my device is **Apple_71:16:27 (9c:3e:53:71:16:27)**. The destination, which is a remote device is **JuniperNetwo_f6:b6:a0 (b0:33:a6:f6:a0)**.

Task: 8



The filter formula “dns && ip.addr == <your IP address> is used in Wireshark analyze just the DNS activity of our own device. It helps us to see which domains our computer is trying to access, and what IP addresses it gets in return.

In this screenshot, the formula “dns && ip.addr == 10.0.110.91” filters the results to display only DNS traffic that involves the IP address 10.0.110.91 (my device).