

Analyzing File Upload with Wireshark

Prerequisites

Install Wireshark from [Wireshark's official website]([Wireshark · Go Deep](#))

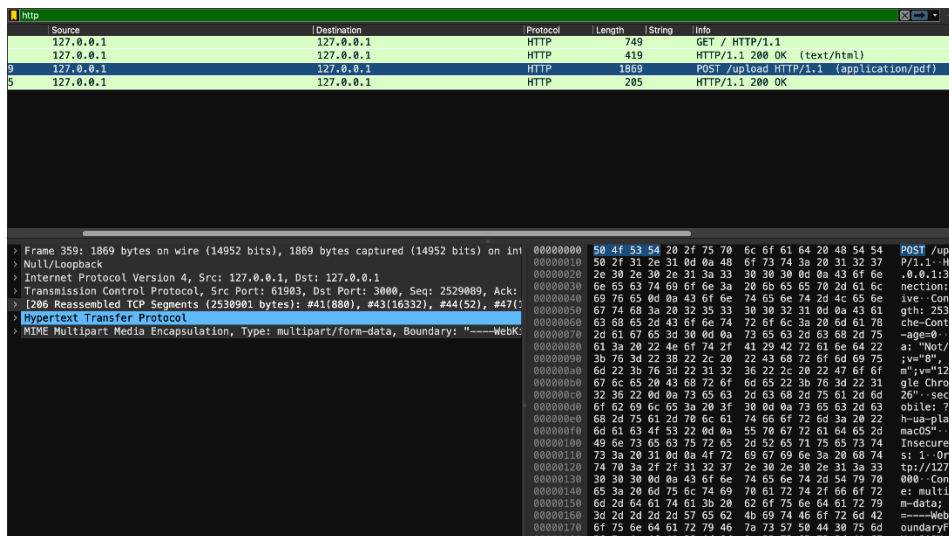
Steps to Capture File Upload

1. Start Wireshark and Capture Traffic

- Open Wireshark.
- Select the network interface that the file upload will occur over.(if you have started the server in your own machine then select the loopback interface)
- Start capturing traffic by clicking the blue shark fin icon.

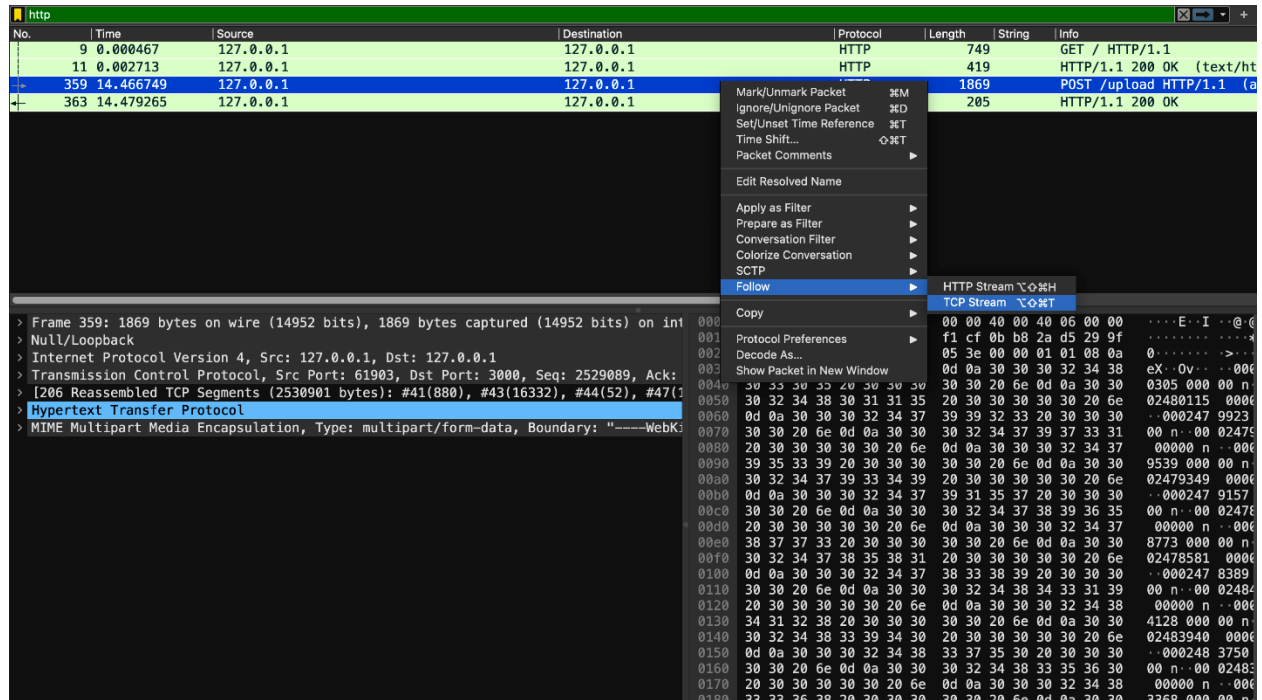
2. Filter HTTP Traffic

- In the filter bar, type 'http' to filter HTTP traffic.
- Press Enter to apply the filter.



3. Locate the File Upload Request

- Identify the HTTP POST request that initiates the file upload.
- The POST request will likely have a large payload if you are uploading a file.



4. Follow the TCP Stream

- Right click on the POST request packet.
- Select "Follow" > "TCP Stream".

```
00000000 50 4f 53 54 20 2f 75 70 6c 6f 61 64 20 48 54 54 POST /up load HTT
00000010 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 32 37 P/1.1..H ost: 127
00000020 2e 30 2e 30 2e 31 3a 33 30 30 30 0d 0a 43 6f 6e .0.0.1:3 000..Con
00000030 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c nection: keep-al
00000040 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e ive..Con tent-Len
00000050 67 74 68 3a 20 32 35 33 30 30 32 31 0d 0a 43 61 gth: 253 0021..Ca
00000060 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 che-Cont rol: max
00000070 2d 61 67 65 34 30 0d 0a 73 65 63 2d 63 68 2d 75 -age=0., sec-ch-u
00000080 61 3a 20 22 4e 6f 74 2f 41 29 42 72 61 6e 64 22 a: "Not/ A)Brand"
00000090 3b 76 3d 22 38 22 2c 20 22 43 68 72 6f 6d 69 75 ;v="8", "Chromiu
000000a0 6d 22 3b 76 3d 22 31 32 36 22 2c 20 22 47 6f 6f m";v="12 6", "Goo
000000b0 67 6c 65 20 43 68 72 6f 6d 65 22 3b 76 3d 22 31 gle Chro me";v="1
000000c0 32 36 22 0d 0a 73 65 63 2d 63 68 2d 75 61 2d 6d 26"..sec -ch-ua-m
000000d0 6f 62 69 6c 65 3a 20 3f 30 0d 0a 73 65 63 2d 63 obile: ? 0..sec-c
000000e0 68 2d 75 61 2d 70 6c 61 74 66 6f 72 6d 3a 20 22 h-ua-pla tform: "
000000f0 6d 61 63 4f 53 22 0d 0a 55 70 67 72 61 64 65 2d macOS".. Upgrade=
00000100 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 74 Insecure -Request
00000110 73 3a 20 31 0d 0a 4f 72 69 67 69 6e 3a 20 68 74 s: 1..Or igin: ht
00000120 74 70 3a 2f 2f 31 32 37 2e 30 2e 30 2e 31 3a 33 tp://127 .0.0.1:3
00000130 30 30 30 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 000..Con tent-Typ
00000140 65 3a 20 6d 75 6c 74 69 70 61 72 74 2f 66 6f 72 e: multi part/for
00000150 6d 2d 64 61 74 61 3b 20 62 6f 75 6e 64 61 72 79 m-data; boundary
00000160 3d 2d 2d 2d 2d 57 65 62 4b 69 74 46 6f 72 6d 42 =----Web KitFormB
00000170 6f 75 6e 64 61 72 79 46 7a 73 57 50 44 30 75 6d oundaryF z5WPD0um
00000180 56 7a 4c 4f 41 30 4d 0d 0a 55 73 65 72 2d 41 67 VLOA0M..User-Ag
00000190 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 ent: Moz illa/5.0
000001a0 20 28 4d 61 63 69 6e 74 6f 73 68 3b 20 49 6e 74 (Macint osh; Int
000001b0 65 6c 20 4d 61 63 20 4f 53 20 58 20 31 30 5f 31 el Mac O S X 10.1
000001c0 35 5f 37 29 20 41 70 70 6c 65 57 65 62 4b 69 74 5.7) App leWebKit
000001d0 2f 35 33 37 2e 33 36 20 28 4b 48 54 4d 4c 2c 20 /537.36 (KHTML,
000001e0 6c 69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72 6f like Gec ko) Chro
000001f0 6d 65 2f 31 32 36 2e 30 2e 30 2e 30 20 53 61 66 me/126.0 .0.0 Saf
00000200 61 72 69 2f 35 33 37 2e 33 36 0d 0a 41 63 63 65 ari/537. 36..Acce
00000210 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 61 70 pt: text /html,ap
00000220 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b plicatio n/xhtml+
00000230 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f xml,appl ication/
00000240 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65 2f xml;q=0. 9,image/
00000250 61 76 69 66 2c 69 6d 61 67 65 2f 77 65 62 70 2c avif,ima ge/webp,
00000260 69 6d 61 67 65 2f 61 70 6e 67 2c 2a 2f 2a 3b 71 image/ap ng,*/*;q
00000270 3d 30 2e 38 2c 61 70 70 6c 69 63 61 74 69 6f 6e =0.8,app lication
00000280 2f 73 69 67 6e 65 64 2d 65 78 63 68 61 6e 67 65 /signed- exchange
00000290 3b 76 3d 62 33 3b 71 3d 30 2e 37 0d 0a 53 65 63 ;v=b3;q= 0.7..Sec
000002a0 2d 46 65 74 63 68 2d 53 69 74 65 3a 20 73 61 6d -Fetch-S ite: sam
000002b0 65 2d 6f 72 69 67 69 6e 0d 0a 53 65 63 2d 46 65 e-origin ..Sec-Fe
```

Packet 41. 205 client pkts, 1 server pkt, 1 turn. Click to select.

Entire conversation (2531 kB) Show data as Hex Dump Stream 1

Find: Find Next

```
00000230 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f xml,appl ication/
00000240 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65 2f xml;q=0. 9,image/
00000250 61 76 69 66 2c 69 6d 61 67 65 2f 77 65 62 70 2c avif,ima ge/webp,
00000260 69 6d 61 67 65 2f 61 70 6e 67 2c 2a 2f 2a 3b 71 image/ap ng,*/*;q
00000270 3d 30 2e 38 2c 61 70 70 6c 69 63 61 74 69 6f 6e =0.8,app lication
00000280 2f 73 69 67 6e 65 64 2d 65 78 63 68 61 6e 67 65 /signed- exchange
00000290 3b 76 3d 62 33 3b 71 3d 30 2e 37 0d 0a 53 65 63 ;v=b3;q= 0.7..Sec
000002a0 2d 46 65 74 63 68 2d 53 69 74 65 3a 20 73 61 6d -Fetch-S ite: sam
000002b0 65 2d 6f 72 69 67 69 6e 0d 0a 53 65 63 2d 46 65 e-origin ..Sec-Fe
```

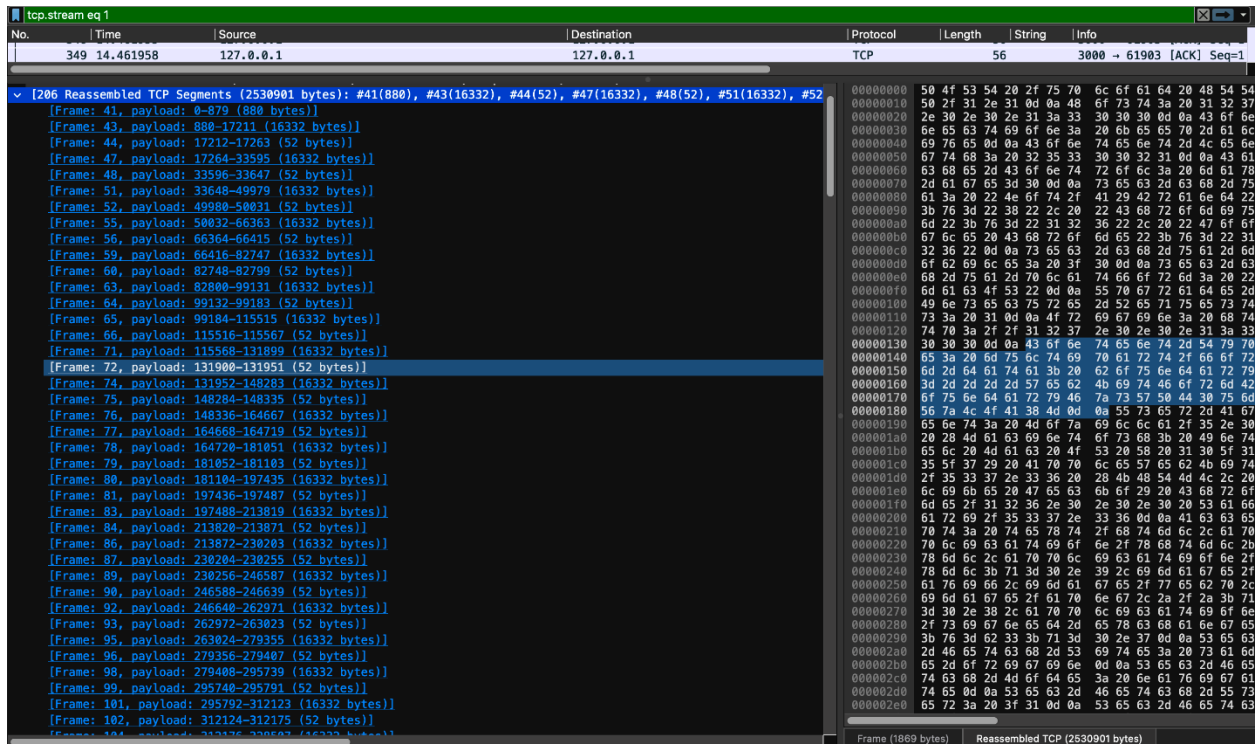
Packet 41. 205 client pkts, 1 server pkt, 1 turn. Click to select.

Entire conversation (2531 kB) Show data as Hex Dump

- This will display all the packets exchanged between the client and the server during the file upload.
- Below we can see that there are altogether 205 client packets and 1 server packet.

5. Analyze the TCP Segments

- In the TCP stream view, you can see all the TCP segments exchanged.
- You can view different TCP segments length and starting and ending bytes transferred from the TCP layer in the window pane.
- Observe the sequence numbers and the length of each segment to understand how the file was divided.



The screenshot displays the Wireshark interface with the TCP Stream View selected. The top pane shows a list of TCP segments, including their sequence numbers and lengths. The middle pane shows the reassembled TCP stream, which is a continuous sequence of bytes. The bottom pane shows the raw packet data in hexadecimal and ASCII. The stream view shows a sequence of bytes starting from 3000 and ending at 61903, with a total length of 31904 bytes. The segments are listed in the top pane, showing their sequence numbers and lengths. The middle pane shows the reassembled TCP stream, which is a continuous sequence of bytes. The bottom pane shows the raw packet data in hexadecimal and ASCII.

- Close the TCP stream view to return to the main Wireshark window.
- Use the filter 'tcp.stream eq <stream_number>' (replace '<stream_number>' with the actual stream number found in the TCP stream view) to isolate the specific TCP stream for the file upload.
- Count the number of segments (packets) in this stream.

Summary

By following these steps, you can capture and analyze how a file is uploaded over the network, including how it is divided into TCP segments. This is useful for performance analysis and troubleshooting.