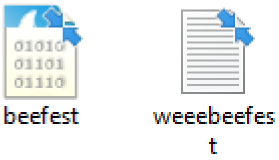
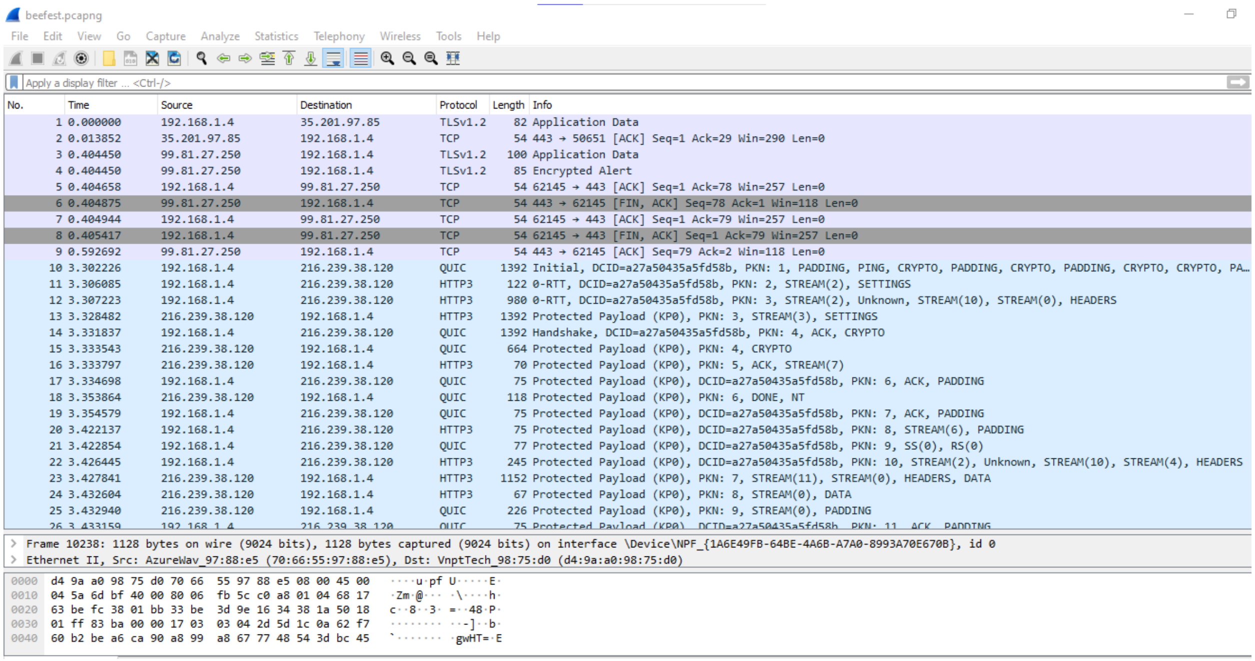
BeeFest 2021 Capture The Flag
Network Forensic Challenge - “Too Sharky”

```
--------------------------------------------------------------------------
I've just sniffed my brother's network unnoticed this morning. And one thing I know,
he's so often use this kinda online-clipboard-website thingy...thing (but i forgot
the name of the website) which I believe that the information is confidential. Can
you help me get the content?
--------------------------------------------------------------------------
```

In this challenge, we were given a packet-capture (*.pcap*) file and text document (*.txt*) file

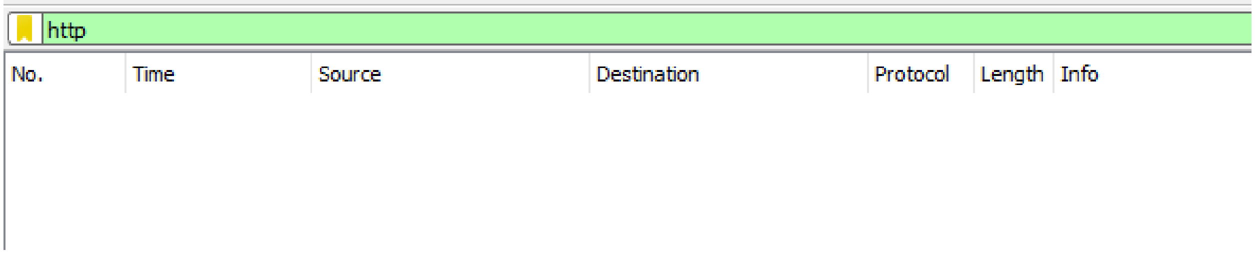


Let’s analyze the packet-capture file with Wireshark.

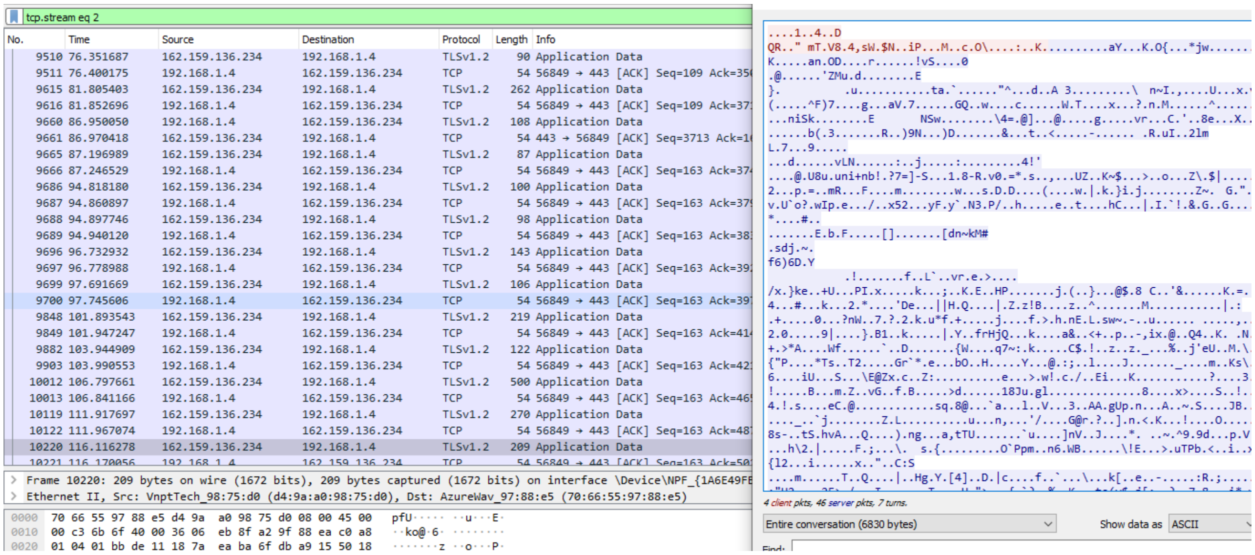

beefest.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time        Source          Destination     Protocol  Length Info
    1 0.000000    192.168.1.4     35.201.97.85    TLSv1.2     82 Application Data
    2 0.013852    35.201.97.85    192.168.1.4     TCP         54 443 → 50651 [ACK] Seq=1 Ack=29 Win=290 Len=0
    3 0.404450    99.81.27.250    192.168.1.4     TLSv1.2    100 Application Data
    4 0.404450    99.81.27.250    192.168.1.4     TLSv1.2     85 Encrypted Alert
    5 0.404658    192.168.1.4     99.81.27.250    TCP         54 62145 → 443 [ACK] Seq=1 Ack=78 Win=257 Len=0
    6 0.404875    99.81.27.250    192.168.1.4     TCP         54 443 → 62145 [FIN, ACK] Seq=78 Ack=1 Win=118 Len=0
    7 0.404944    192.168.1.4     99.81.27.250    TCP         54 62145 → 443 [ACK] Seq=1 Ack=79 Win=257 Len=0
    8 0.405417    192.168.1.4     99.81.27.250    TCP         54 62145 → 443 [FIN, ACK] Seq=1 Ack=79 Win=257 Len=0
    9 0.592692    99.81.27.250    192.168.1.4     TCP         54 443 → 62145 [ACK] Seq=79 Ack=2 Win=118 Len=0
   10 3.302226    192.168.1.4     216.239.38.120  QUIC      1392 Initial, DCID=a27a50435a5fd58b, PKN: 1, PADDING, PING, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, CRYPTO, PA…
   11 3.306085    192.168.1.4     216.239.38.120  HTTP3      122 0-RTT, DCID=a27a50435a5fd58b, PKN: 2, STREAM(2), SETTINGS
   12 3.307223    192.168.1.4     216.239.38.120  HTTP3      980 0-RTT, DCID=a27a50435a5fd58b, PKN: 3, STREAM(2), Unknown, STREAM(10), STREAM(0), HEADERS
   13 3.328482    216.239.38.120  192.168.1.4     HTTP3     1392 Protected Payload (KP0), PKN: 3, STREAM(3), SETTINGS
   14 3.331837    192.168.1.4     216.239.38.120  QUIC      1392 Handshake, DCID=a27a50435a5fd58b, PKN: 4, ACK, CRYPTO
   15 3.333543    216.239.38.120  192.168.1.4     QUIC       664 Protected Payload (KP0), PKN: 4, CRYPTO
   16 3.333797    216.239.38.120  192.168.1.4     HTTP3       70 Protected Payload (KP0), PKN: 5, ACK, STREAM(7)
   17 3.334698    192.168.1.4     216.239.38.120  QUIC        75 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 6, ACK, PADDING
   18 3.353864    216.239.38.120  192.168.1.4     QUIC       118 Protected Payload (KP0), PKN: 6, DONE, NT
   19 3.354579    192.168.1.4     216.239.38.120  QUIC        75 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 7, ACK, PADDING
   20 3.422137    192.168.1.4     216.239.38.120  HTTP3       75 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 8, STREAM(6), PADDING
   21 3.422854    192.168.1.4     216.239.38.120  QUIC        77 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 9, SS(0), RS(0)
   22 3.426445    192.168.1.4     216.239.38.120  HTTP3      245 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 10, STREAM(2), Unknown, STREAM(10), STREAM(4), HEADERS
   23 3.427841    216.239.38.120  192.168.1.4     HTTP3     1152 Protected Payload (KP0), PKN: 7, STREAM(11), STREAM(0), HEADERS, DATA
   24 3.432604    216.239.38.120  192.168.1.4     HTTP3       67 Protected Payload (KP0), PKN: 8, STREAM(0), DATA
   25 3.432940    216.239.38.120  192.168.1.4     QUIC       226 Protected Payload (KP0), PKN: 9, STREAM(0), PADDING
   26 3.433159    192.168.1.4     216.239.38.120  QUIC        75 Protected Payload (KP0), DCID=a27a50435a5fd58b, PKN: 11, ACK, PADDING

> Frame 10238: 1128 bytes on wire (9024 bits), 1128 bytes captured (9024 bits) on interface \Device\NPF_{1A6E49FB-64BE-4A6B-A7A0-8993A70E670B}, id 0
> Ethernet II, Src: AzureWav_97:88:e5 (70:66:55:97:88:e5), Dst: VnptTech_98:75:d0 (d4:9a:a0:98:75:d0)

0000  d4 9a a0 98 75 d0 70 66  55 97 88 e5 08 00 45 00   ····u·pf U·····E·
0010  04 5a 6d bf 40 00 80 06  fb 5c c0 a8 01 04 68 17   ·Zm·@··· ·\····h·
0020  63 be fc 38 01 bb 33 be  3d 9e 16 34 38 1a 50 18   c··8··3· =··48·P·
0030  01 ff 83 ba 00 00 17 03  03 04 2d 5d 1c 0a 62 f7   ········ ··-]··b·
0040  60 b2 be a6 ca 90 a8 99  a8 67 77 48 54 3d bc 45   `······· ·gwHT=·E
```

After a brief analysis, it turned out that there was no HTTP protocol and also all the content is not readable from the client to the server and vice versa.



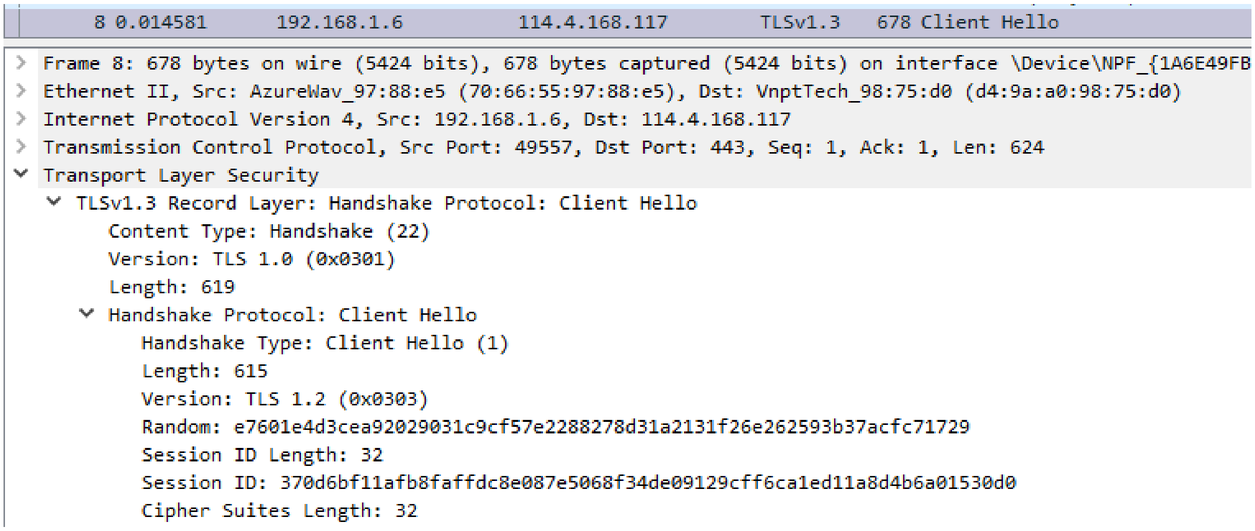
```
http
No.    Time    Source    Destination    Protocol    Length    Info
```



```
tcp.stream eq 2
No.    Time        Source           Destination      Protocol Length Info
 9510 76.351687   162.159.136.234  192.168.1.4      TLSv1.2    90 Application Data
 9511 76.400175   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=109 Ack=35
 9615 81.805403   162.159.136.234  192.168.1.4      TLSv1.2   262 Application Data
 9616 81.852696   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=109 Ack=37
 9660 86.950050   192.168.1.4      162.159.136.234  TLSv1.2   108 Application Data
 9661 86.970418   162.159.136.234  192.168.1.4      TCP        54 443 → 56849 [ACK] Seq=3713 Ack=1
 9665 87.196989   162.159.136.234  192.168.1.4      TLSv1.2    87 Application Data
 9666 87.246529   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=37
 9686 94.818180   162.159.136.234  192.168.1.4      TLSv1.2   100 Application Data
 9687 94.860897   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=37
 9688 94.897746   162.159.136.234  192.168.1.4      TLSv1.2    98 Application Data
 9689 94.940120   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=38
 9696 96.732932   162.159.136.234  192.168.1.4      TLSv1.2   143 Application Data
 9697 96.778988   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=39
 9699 97.691669   162.159.136.234  192.168.1.4      TLSv1.2   106 Application Data
 9700 97.745606   192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=39
 9848 101.893543  162.159.136.234  192.168.1.4      TLSv1.2   219 Application Data
 9849 101.947247  192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=41
 9882 103.944909  162.159.136.234  192.168.1.4      TLSv1.2   122 Application Data
 9903 103.990553  192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=42
10012 106.797661  162.159.136.234  192.168.1.4      TLSv1.2   500 Application Data
10013 106.841166  192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=46
10119 111.917697  162.159.136.234  192.168.1.4      TLSv1.2   270 Application Data
10122 111.967074  192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=48
10220 116.116278  162.159.136.234  192.168.1.4      TLSv1.2   209 Application Data
10221 116.170056  192.168.1.4      162.159.136.234  TCP        54 56849 → 443 [ACK] Seq=163 Ack=50

> Frame 10220: 209 bytes on wire (1672 bits), 209 bytes captured (1672 bits) on interface \Device\NPF_{1A6E49FE
> Ethernet II, Src: VnptTech_98:75:d0 (d4:9a:a0:98:75:d0), Dst: AzureWav_97:88:e5 (70:66:55:97:88:e5)

0000  70 66 55 97 88 e5 d4 9a  a0 98 75 d0 08 00 45 00   pfU·····  ··u···E·
0010  00 c3 6b 6f 40 00 36 06  eb 8f a2 9f 88 ea c0 a8   ··ko@·6·  ········
0020  01 04 01 bb de 11 18 7a  ea ba 6f db a9 15 50 18   ·······z  ··o···P·
```

```
....1..4..D
QR.." mT.V8.4,sW.$N..iP...M..c.O\....:..K..........aY...K.O{...*jw......
K.....an.OD....r......!vS....0
.@......'ZMu.d........E
}.          .u............ta.`......"^...d..A 3.........\  n~I.,....U...x.
(.....^F)7....g...aV.7......GQ..w....c......W.T....x...?.n.M......^.....
...niSk........E       NSw.........\4=.@]...@.....g.....vr...C.'..8e...X..
......b(.3.......R..)9N...)D.......&...t..<.....-...... .R.uI..2lm
L.7...9.....
...d......vLN......:..j.....:.........4!'
....@.U8u.uni+nb!.?7=]-S...1.8-R.v0.=*.s..,...UZ..K~$...>..o...Z\.$|.....
2...p.=..mR...F....m........w...s.D.D....(....w.|.k.}i.j........Z~.  G.".
v.U`o?.wIp.e.../..x52...yF.y`.N3.P/..h.....e..t....hC...|.I.`!.&.G..G...
*....#..
.......E.b.F.....[].......[dn~kM#
.sdj.~.
f6)6D.Y
            .!.......f..L`..vr.e.>....
/x.}ke..+U...PI.x.....k...;..K.E..HP.......j.(..}...@$.8 C..'&......K.=.
4...#...k...2.*....'De...||H.Q....|.Z.z!B....z..^.......M...........|.:
.+.....0...?nW..7.?.2.k.u*f.+.....j....f.>.h.nE.L.sw~.-..u......  .....,..
2.0.....9|....}.B1..k.....|.Y..frHjQ...k....a&..<+..p..-,ix.@..Q4..K.  .N.
+.>*A....Wf......`..D.......{W....q7~:.k.....C$.!..z..z._...%..j'eU..M.\.
{"P....*Ts..T2.....Gr`*.e...bO..H.....Y...@.:;..l....J......_....m..Ks\.
6....iU...S...\E@Zx.c..Z:..........e...>.w!.c./..Ei...K...........?....3.
!.....B...m.Z..vG..f.B.....>d......18Ju.gl..............8....x>....S..!..
4.!.s....eC.@............sq.8@...`a...l..V...3..AA.gUp.n...A..~.S....JB..
...._..`j........Z.L..........u...n,...'/....G@r.?..].n.<.K...!....O.....
8s-..tS.hvA...Q....).ng...a,tTU.......`u....]nV..J....*.  ..~.^9.9d...p.V
...h\2.|.....F.;...\.  s.{.........O`Ppm..n6.WB......\!E...>.uTPb.<..i..
{l2...i......x..."..C:S
....m......T..Q....|..Hg.Y.[4]..D.|c....f..`...\...k[..e..-.....:R.;.....

4 client pkts, 46 server pkts, 7 turns.
Entire conversation (6830 bytes)        Show data as  ASCII
```

Well, it can be seen here that there is an encrypted conversation between the client and the server, namely the client talking to the server with the HTTPS or HTTP Secure website protocol with TLSv1.3 encryption.



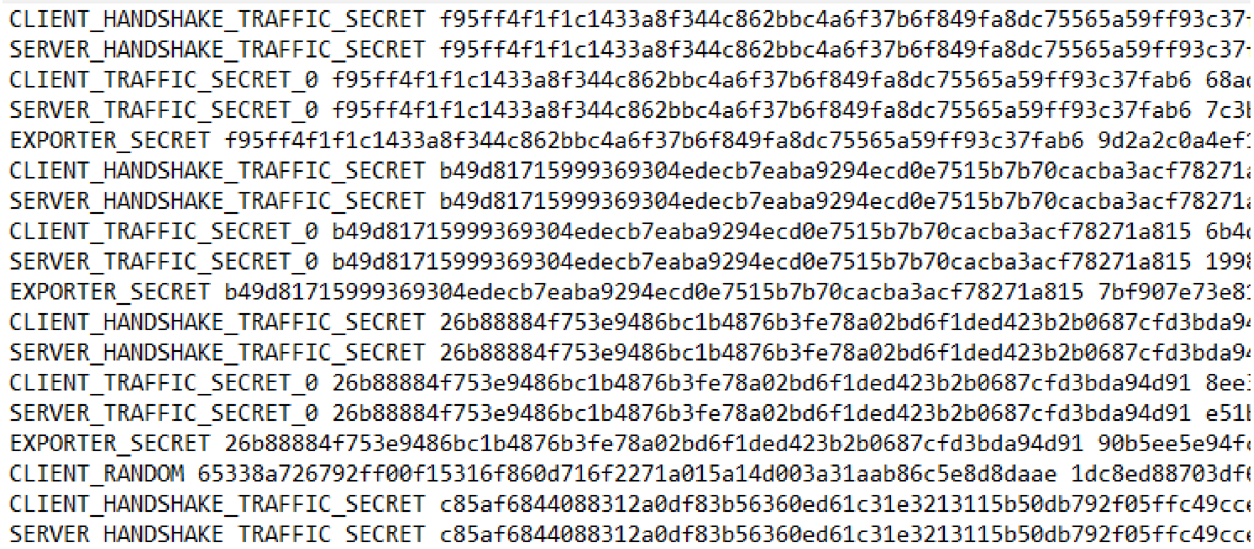
```
     8 0.014581      192.168.1.6        114.4.168.117       TLSv1.3    678 Client Hello

> Frame 8: 678 bytes on wire (5424 bits), 678 bytes captured (5424 bits) on interface \Device\NPF_{1A6E49FB
> Ethernet II, Src: AzureWav_97:88:e5 (70:66:55:97:88:e5), Dst: VnptTech_98:75:d0 (d4:9a:a0:98:75:d0)
> Internet Protocol Version 4, Src: 192.168.1.6, Dst: 114.4.168.117
> Transmission Control Protocol, Src Port: 49557, Dst Port: 443, Seq: 1, Ack: 1, Len: 624
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 619
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 615
        Version: TLS 1.2 (0x0303)
        Random: e7601e4d3cea92029031c9cf57e2288278d31a2131f26e262593b37acfc71729
        Session ID Length: 32
        Session ID: 370d6bf11afb8faffdc8e087e5068f34de09129cff6ca1ed11a8d4b6a01530d0
        Cipher Suites Length: 32
```

Also the encrypted conversation comes from IP Address 192.168.1.4

Gathered Information

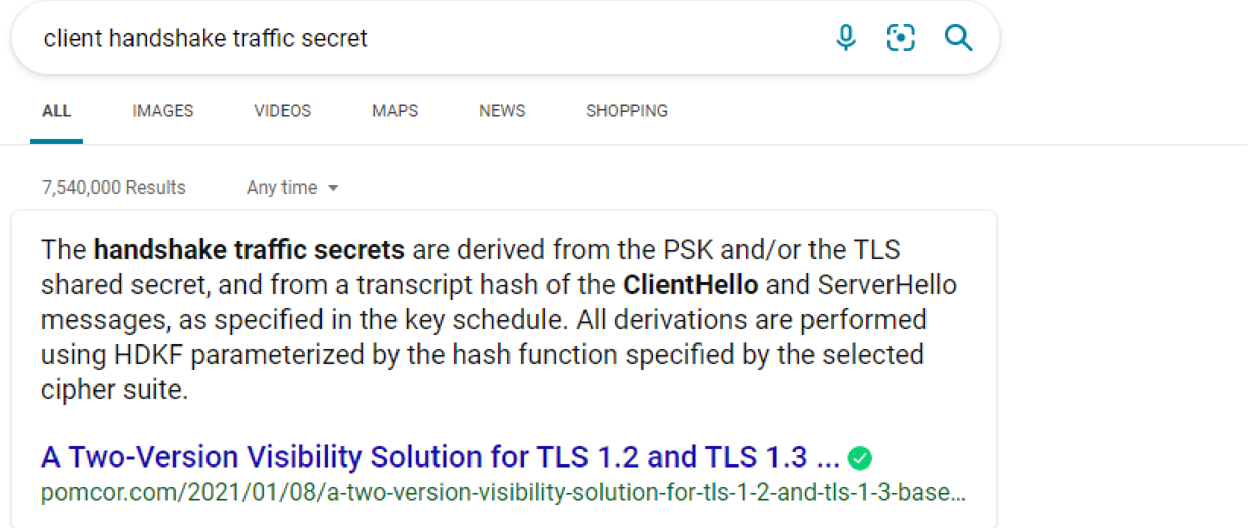
So far, the information we have got is:
>> Suspected IP Address = 192.168.1.4
>> Cryptographic Protocol / Its Traffic Encryption = TLSv1.3

Move on to the second file, which is the text document file.
In that file, contains abstract character, except its prefix



```
CLIENT_HANDSHAKE_TRAFFIC_SECRET f95ff4f1f1c1433a8f344c862bbc4a6f37b6f849fa8dc75565a59ff93c37
SERVER_HANDSHAKE_TRAFFIC_SECRET f95ff4f1f1c1433a8f344c862bbc4a6f37b6f849fa8dc75565a59ff93c37
CLIENT_TRAFFIC_SECRET_0 f95ff4f1f1c1433a8f344c862bbc4a6f37b6f849fa8dc75565a59ff93c37fab6 68a
SERVER_TRAFFIC_SECRET_0 f95ff4f1f1c1433a8f344c862bbc4a6f37b6f849fa8dc75565a59ff93c37fab6 7c3
EXPORTER_SECRET f95ff4f1f1c1433a8f344c862bbc4a6f37b6f849fa8dc75565a59ff93c37fab6 9d2a2c0a4ef
CLIENT_HANDSHAKE_TRAFFIC_SECRET b49d81715999369304edecb7eaba9294ecd0e7515b7b70cacba3acf78271
SERVER_HANDSHAKE_TRAFFIC_SECRET b49d81715999369304edecb7eaba9294ecd0e7515b7b70cacba3acf78271
CLIENT_TRAFFIC_SECRET_0 b49d81715999369304edecb7eaba9294ecd0e7515b7b70cacba3acf78271a815 6b4
SERVER_TRAFFIC_SECRET_0 b49d81715999369304edecb7eaba9294ecd0e7515b7b70cacba3acf78271a815 199
EXPORTER_SECRET b49d81715999369304edecb7eaba9294ecd0e7515b7b70cacba3acf78271a815 7bf907e73e8
CLIENT_HANDSHAKE_TRAFFIC_SECRET 26b88884f753e9486bc1b4876b3fe78a02bd6f1ded423b2b0687cfd3bda9
SERVER_HANDSHAKE_TRAFFIC_SECRET 26b88884f753e9486bc1b4876b3fe78a02bd6f1ded423b2b0687cfd3bda9
CLIENT_TRAFFIC_SECRET_0 26b88884f753e9486bc1b4876b3fe78a02bd6f1ded423b2b0687cfd3bda94d91 8ee
SERVER_TRAFFIC_SECRET_0 26b88884f753e9486bc1b4876b3fe78a02bd6f1ded423b2b0687cfd3bda94d91 e51
EXPORTER_SECRET 26b88884f753e9486bc1b4876b3fe78a02bd6f1ded423b2b0687cfd3bda94d91 90b5ee5e94f
CLIENT_RANDOM 65338a726792ff00f15316f860d716f2271a015a14d003a31aab86c5e8d8daae 1dc8ed88703df
CLIENT_HANDSHAKE_TRAFFIC_SECRET c85af6844088312a0df83b56360ed61c31e3213115b50db792f05ffc49cc
SERVER HANDSHAKE TRAFFIC SECRET c85af6844088312a0df83b56360ed61c31e3213115b50db792f05ffc49cc
```

If we do not recognize that in first glance, we can check that on Google.
The keyword search is like down below :



The **handshake traffic secrets** are derived from the PSK and/or the TLS shared secret, and from a transcript hash of the **ClientHello** and ServerHello messages, as specified in the key schedule. All derivations are performed using HDKF parameterized by the hash function specified by the selected cipher suite.

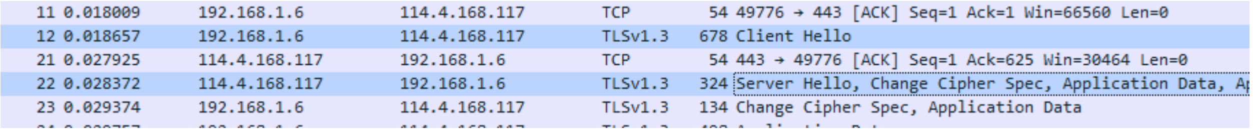
A Two-Version Visibility Solution for TLS 1.2 and TLS 1.3 ...
pomcor.com/2021/01/08/a-two-version-visibility-solution-for-tls-1-2-and-tls-1-3-base...

After searching, we also get to know a thing about **Client Hello** or **Server Hello**

And why is the suspected IP Address is 192.168.1.4?

As already stated, Client Hello means from Client conveying a message or request to the server or an endpoint, while Server Hello means sending a message back to the Client.

It can be seen from picture down below :



```
11 0.018009    192.168.1.6      114.4.168.117    TCP        54 49776 → 443 [ACK] Seq=1 Ack=1 Win=66560 Len=0
12 0.018657    192.168.1.6      114.4.168.117    TLSv1.3   678 Client Hello
21 0.027925    114.4.168.117    192.168.1.6      TCP        54 443 → 49776 [ACK] Seq=1 Ack=625 Win=30464 Len=0
22 0.028372    114.4.168.117    192.168.1.6      TLSv1.3   324 Server Hello, Change Cipher Spec, Application Data, Ap
23 0.029374    192.168.1.6      114.4.168.117    TLSv1.3   134 Change Cipher Spec, Application Data
```

The IP Address on the left is the source, and on the right is the destination.
With the Client Hello at source → 192.168.1.4

Now, all we have to do is **decrypt the HTTPS traffic.**

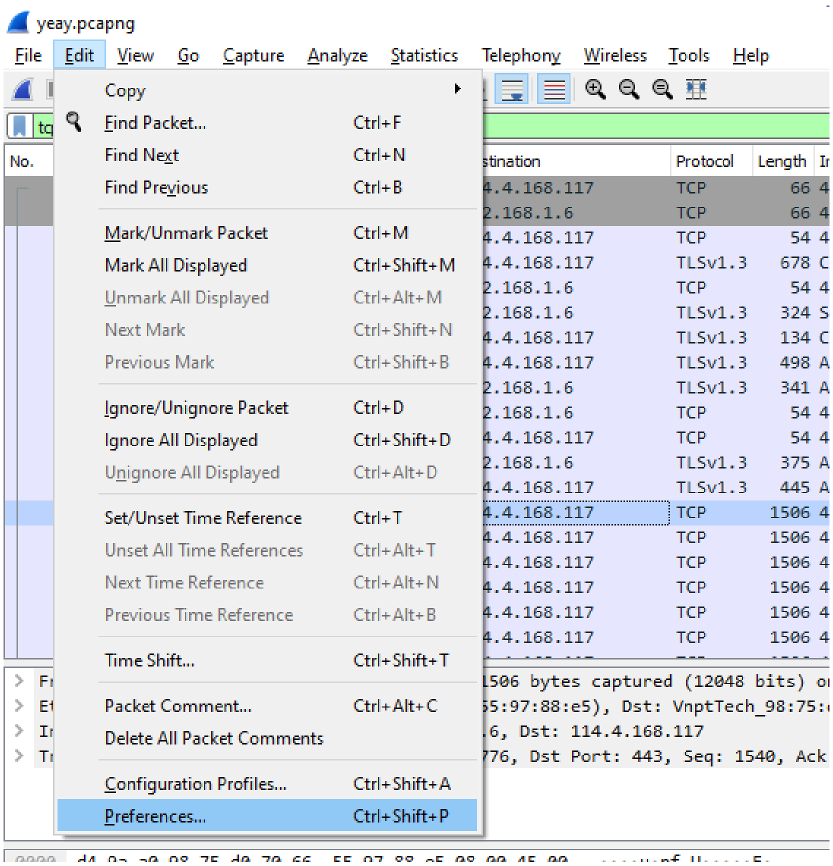
(Check out this link down below on how to decrypt HTTPS Traffic)
[Wireshark Tutorial: Decrypting HTTPS Traffic (Includes SSL and TLS) (paloaltonetworks.com)]

HTTPS Decryption

From the previous reference link, the document file that we have contains the SSL Key Log that will be used to decrypt the TLS Encryption. Without an SSL Key Log made during the packet capture process, HTTPS traffic that is successfully stamped/captured will be useless, except only to see the IP Address, port, how many packets are in each send-data, and so on.

The information that we want to find is the data that moves from the Client to the Server and vice versa contained in the header and body of the website.
Here's how to decrypt TLS with Wireshark:

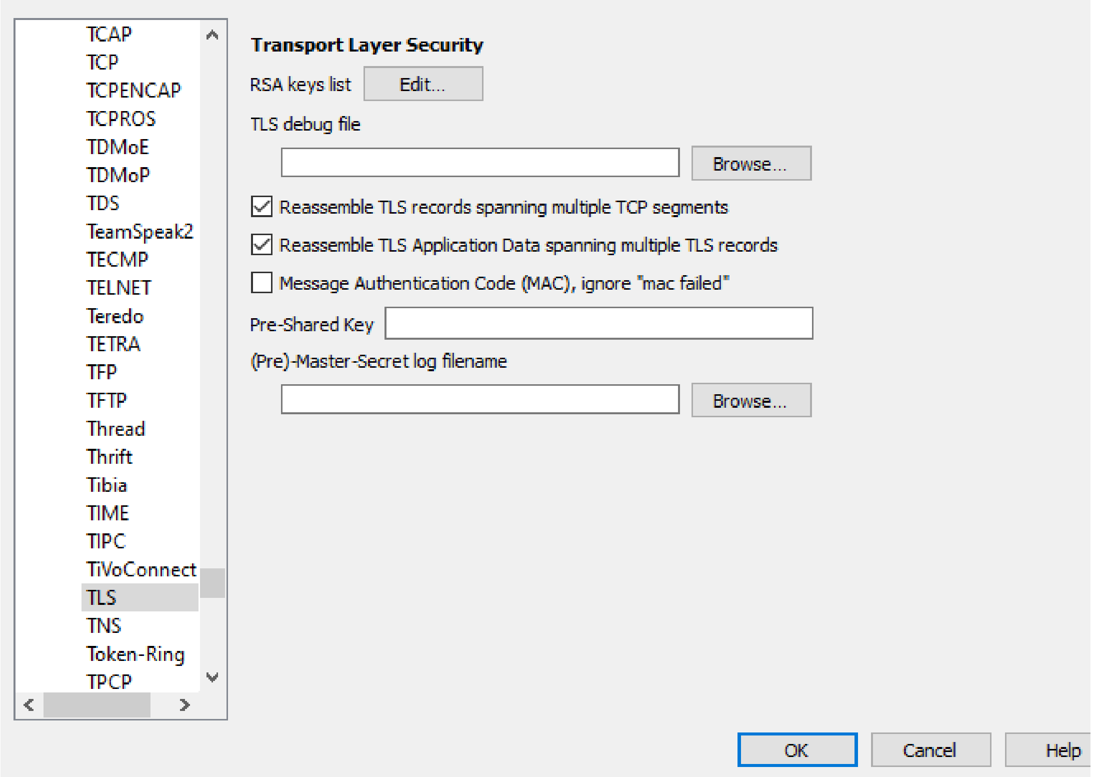
1. Go to Edit → References


yeay.pcapng
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles... Ctrl+Shift+A
Preferences... Ctrl+Shift+P
No.
tc

```
stination       Protocol  Length  I
4.4.168.117     TCP           66  4
2.168.1.6       TCP           66  4
4.4.168.117     TCP           54  4
4.4.168.117     TLSv1.3      678  C
2.168.1.6       TCP           54  4
2.168.1.6       TLSv1.3      324  S
4.4.168.117     TLSv1.3      134  C
4.4.168.117     TLSv1.3      498  A
2.168.1.6       TLSv1.3      341  A
2.168.1.6       TCP           54  4
4.4.168.117     TCP           54  4
2.168.1.6       TLSv1.3      375  A
4.4.168.117     TLSv1.3      445  A
4.4.168.117     TCP         1506  4
4.4.168.117     TCP         1506  4
4.4.168.117     TCP         1506  4
4.4.168.117     TCP         1506  4
4.4.168.117     TCP         1506  4
4.4.168.117     TCP         1506  4
```

```
> Fr  1506 bytes captured (12048 bits) o
> Et  55:97:88:e5), Dst: VnptTech_98:75:
> Ir  .6, Dst: 114.4.168.117
> Tr  776, Dst Port: 443, Seq: 1540, Ack
```

2. Next, on Protocols click the drop-down button and select TLS



Transport Layer Security

RSA keys list

TLS debug file

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared Key

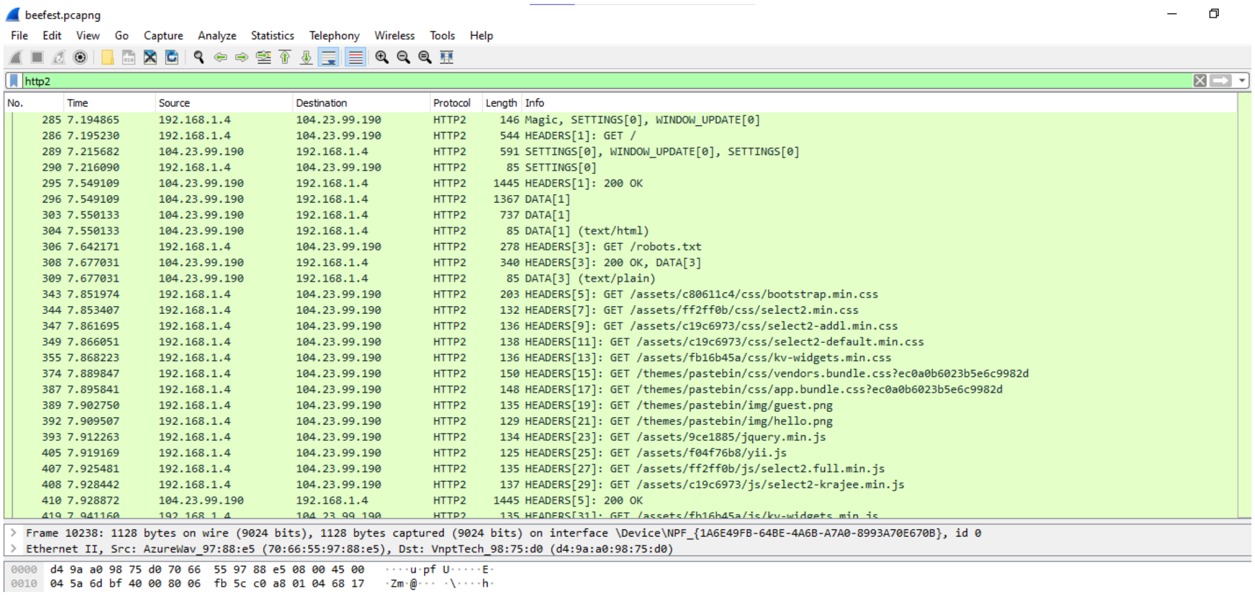
(Pre)-Master-Secret log filename

3. Then click Browse and enter the SSL Key Log in the section (Pre)-Master Secret log filename, which is the following text file:

weeebeefest

4. Click OK after finished.

And then we can scroll around and find a packet with a green color, meaning that it is the HTTP from decrypting the HTTPS traffic. Now we can see the content that was unreadable before.

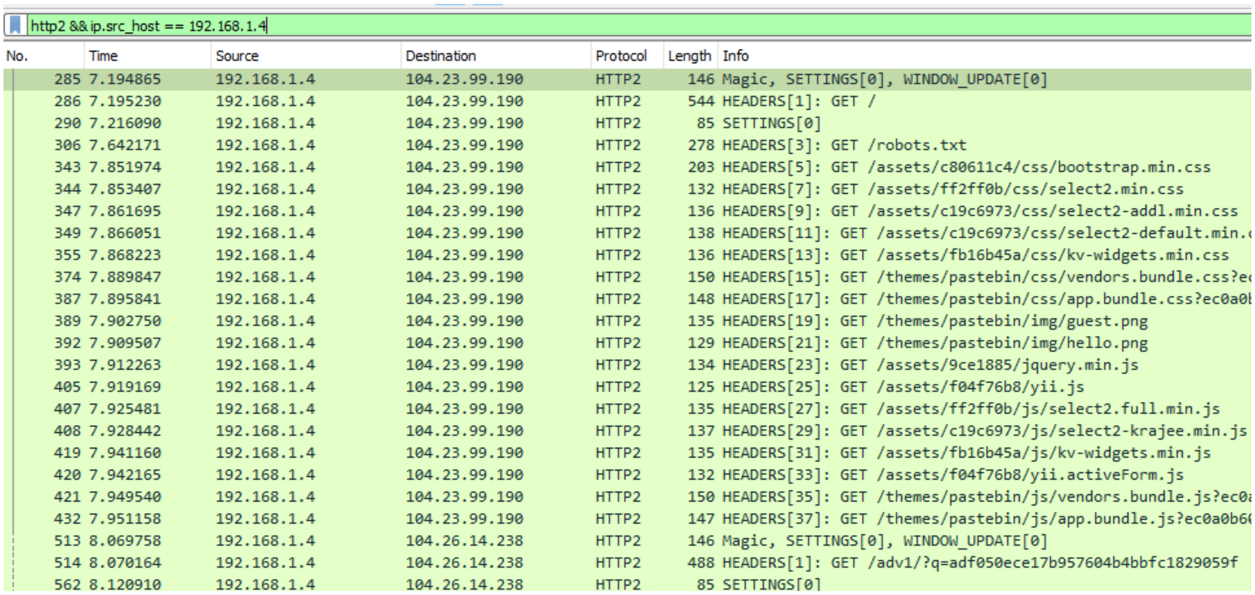


beefest.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http2

No.   Time      Source          Destination     Protocol  Length Info
  285 7.194865  192.168.1.4     104.23.99.190   HTTP2       146 Magic, SETTINGS[0], WINDOW_UPDATE[0]
  286 7.195230  192.168.1.4     104.23.99.190   HTTP2       544 HEADERS[1]: GET /
  289 7.215682  104.23.99.190   192.168.1.4     HTTP2       591 SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
  290 7.216090  192.168.1.4     104.23.99.190   HTTP2        85 SETTINGS[0]
  295 7.549109  104.23.99.190   192.168.1.4     HTTP2      1445 HEADERS[1]: 200 OK
  296 7.549109  104.23.99.190   192.168.1.4     HTTP2      1367 DATA[1]
  303 7.550133  104.23.99.190   192.168.1.4     HTTP2       737 DATA[1]
  304 7.550133  104.23.99.190   192.168.1.4     HTTP2        85 DATA[1] (text/html)
  306 7.642171  192.168.1.4     104.23.99.190   HTTP2       278 HEADERS[3]: GET /robots.txt
  308 7.677031  104.23.99.190   192.168.1.4     HTTP2       340 HEADERS[3]: 200 OK, DATA[3]
  309 7.677031  104.23.99.190   192.168.1.4     HTTP2        85 DATA[3] (text/plain)
  343 7.851974  192.168.1.4     104.23.99.190   HTTP2       203 HEADERS[5]: GET /assets/c80611c4/css/bootstrap.min.css
  344 7.853407  192.168.1.4     104.23.99.190   HTTP2       132 HEADERS[7]: GET /assets/ff2ff0b/css/select2.min.css
  347 7.861695  192.168.1.4     104.23.99.190   HTTP2       136 HEADERS[9]: GET /assets/c19c6973/css/select2-addl.min.css
  349 7.866051  192.168.1.4     104.23.99.190   HTTP2       138 HEADERS[11]: GET /assets/c19c6973/css/select2-default.min.css
  355 7.868223  192.168.1.4     104.23.99.190   HTTP2       136 HEADERS[13]: GET /assets/fb16b45a/css/kv-widgets.min.css
  374 7.889847  192.168.1.4     104.23.99.190   HTTP2       150 HEADERS[15]: GET /themes/pastebin/css/vendors.bundle.css?ec0a0b6023b5e6c9982d
  387 7.895841  192.168.1.4     104.23.99.190   HTTP2       148 HEADERS[17]: GET /themes/pastebin/css/app.bundle.css?ec0a0b6023b5e6c9982d
  389 7.902750  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[19]: GET /themes/pastebin/img/guest.png
  392 7.909507  192.168.1.4     104.23.99.190   HTTP2       129 HEADERS[21]: GET /themes/pastebin/img/hello.png
  393 7.912263  192.168.1.4     104.23.99.190   HTTP2       134 HEADERS[23]: GET /assets/9ce1885/jquery.min.js
  405 7.919169  192.168.1.4     104.23.99.190   HTTP2       125 HEADERS[25]: GET /assets/f04f76b8/yii.js
  407 7.925481  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[27]: GET /assets/ff2ff0b/js/select2.full.min.js
  408 7.928442  192.168.1.4     104.23.99.190   HTTP2       137 HEADERS[29]: GET /assets/c19c6973/js/select2-krajee.min.js
  410 7.928872  104.23.99.190   192.168.1.4     HTTP2      1445 HEADERS[5]: 200 OK
  419 7.941160  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[31]: GET /assets/fb16b45a/js/kv-widgets.min.js

> Frame 10238: 1128 bytes on wire (9024 bits), 1128 bytes captured (9024 bits) on interface \Device\NPF_{1A6E49FB-64BE-4A6B-A7A0-8993A70E670B}, id 0
> Ethernet II, Src: AzureWav_97:88:e5 (70:66:55:97:88:e5), Dst: VnptTech_98:75:d0 (d4:9a:a0:98:75:d0)

0000  d4 9a a0 98 75 d0 70 66  55 97 88 e5 08 00 45 00   ····u·pf U·····E·
0010  04 5a 6d bf 40 00 80 06  fb 5c c0 a8 01 04 68 17   ·Zm·@···  ·\····h·
```

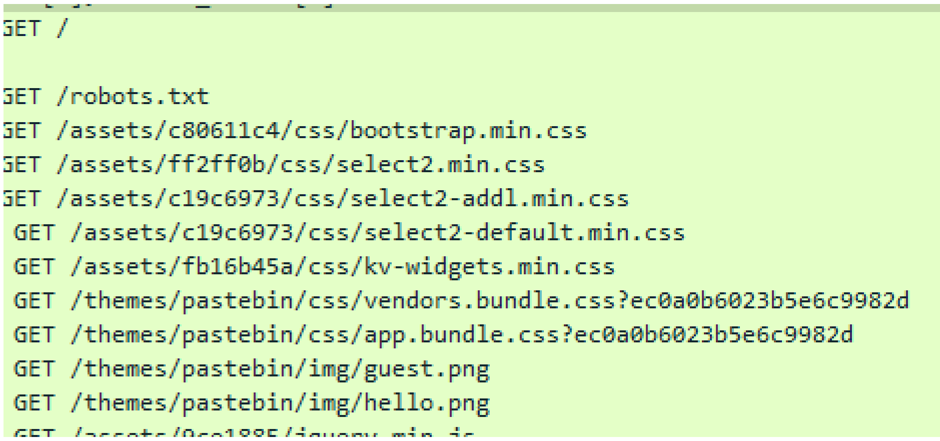
Then, we can use filters to speed up the search with the following syntax:
>> **http2 &&amp ip.src_host == 192.168.1.4**
The syntax means we will only list packets with the HTTP2 protocol and will place IP Address 192.168.1.6 in the source column.

This is the result :



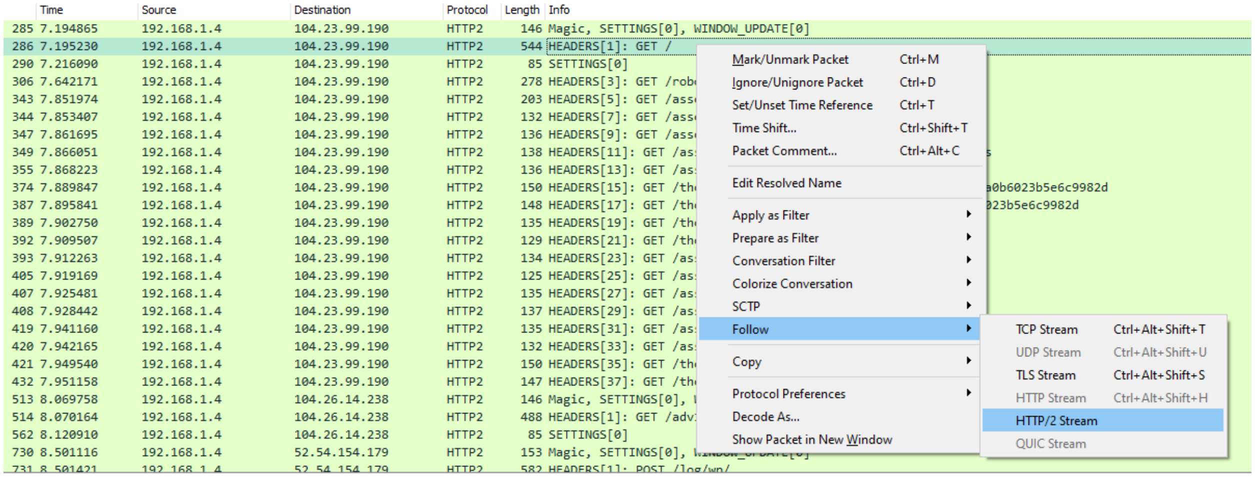
```
http2 && ip.src_host == 192.168.1.4

No.   Time      Source          Destination     Protocol  Length Info
  285 7.194865  192.168.1.4     104.23.99.190   HTTP2       146 Magic, SETTINGS[0], WINDOW_UPDATE[0]
  286 7.195230  192.168.1.4     104.23.99.190   HTTP2       544 HEADERS[1]: GET /
  290 7.216090  192.168.1.4     104.23.99.190   HTTP2        85 SETTINGS[0]
  306 7.642171  192.168.1.4     104.23.99.190   HTTP2       278 HEADERS[3]: GET /robots.txt
  343 7.851974  192.168.1.4     104.23.99.190   HTTP2       203 HEADERS[5]: GET /assets/c80611c4/css/bootstrap.min.css
  344 7.853407  192.168.1.4     104.23.99.190   HTTP2       132 HEADERS[7]: GET /assets/ff2ff0b/css/select2.min.css
  347 7.861695  192.168.1.4     104.23.99.190   HTTP2       136 HEADERS[9]: GET /assets/c19c6973/css/select2-addl.min.css
  349 7.866051  192.168.1.4     104.23.99.190   HTTP2       138 HEADERS[11]: GET /assets/c19c6973/css/select2-default.min.
  355 7.868223  192.168.1.4     104.23.99.190   HTTP2       136 HEADERS[13]: GET /assets/fb16b45a/css/kv-widgets.min.css
  374 7.889847  192.168.1.4     104.23.99.190   HTTP2       150 HEADERS[15]: GET /themes/pastebin/css/vendors.bundle.css?ec
  387 7.895841  192.168.1.4     104.23.99.190   HTTP2       148 HEADERS[17]: GET /themes/pastebin/css/app.bundle.css?ec0a0b
  389 7.902750  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[19]: GET /themes/pastebin/img/guest.png
  392 7.909507  192.168.1.4     104.23.99.190   HTTP2       129 HEADERS[21]: GET /themes/pastebin/img/hello.png
  393 7.912263  192.168.1.4     104.23.99.190   HTTP2       134 HEADERS[23]: GET /assets/9ce1885/jquery.min.js
  405 7.919169  192.168.1.4     104.23.99.190   HTTP2       125 HEADERS[25]: GET /assets/f04f76b8/yii.js
  407 7.925481  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[27]: GET /assets/ff2ff0b/js/select2.full.min.js
  408 7.928442  192.168.1.4     104.23.99.190   HTTP2       137 HEADERS[29]: GET /assets/c19c6973/js/select2-krajee.min.js
  419 7.941160  192.168.1.4     104.23.99.190   HTTP2       135 HEADERS[31]: GET /assets/fb16b45a/js/kv-widgets.min.js
  420 7.942165  192.168.1.4     104.23.99.190   HTTP2       132 HEADERS[33]: GET /assets/f04f76b8/yii.activeForm.js
  421 7.949540  192.168.1.4     104.23.99.190   HTTP2       150 HEADERS[35]: GET /themes/pastebin/js/vendors.bundle.js?ec0a
  432 7.951158  192.168.1.4     104.23.99.190   HTTP2       147 HEADERS[37]: GET /themes/pastebin/js/app.bundle.js?ec0a0b6
  513 8.069758  192.168.1.4     104.26.14.238   HTTP2       146 Magic, SETTINGS[0], WINDOW_UPDATE[0]
  514 8.070164  192.168.1.4     104.26.14.238   HTTP2       488 HEADERS[1]: GET /adv1/?q=adf050ece17b957604b4bbfc1829059f
  562 8.120910  192.168.1.4     104.26.14.238   HTTP2        85 SETTINGS[0]
```

From here, it can be seen directly that IP Address 192.168.1.4 visited the website pastebin.com



```
GET /

GET /robots.txt
GET /assets/c80611c4/css/bootstrap.min.css
GET /assets/ff2ff0b/css/select2.min.css
GET /assets/c19c6973/css/select2-addl.min.css
 GET /assets/c19c6973/css/select2-default.min.css
 GET /assets/fb16b45a/css/kv-widgets.min.css
 GET /themes/pastebin/css/vendors.bundle.css?ec0a0b6023b5e6c9982d
 GET /themes/pastebin/css/app.bundle.css?ec0a0b6023b5e6c9982d
 GET /themes/pastebin/img/guest.png
 GET /themes/pastebin/img/hello.png
```

Now let's follow the HTTP/2 Stream.

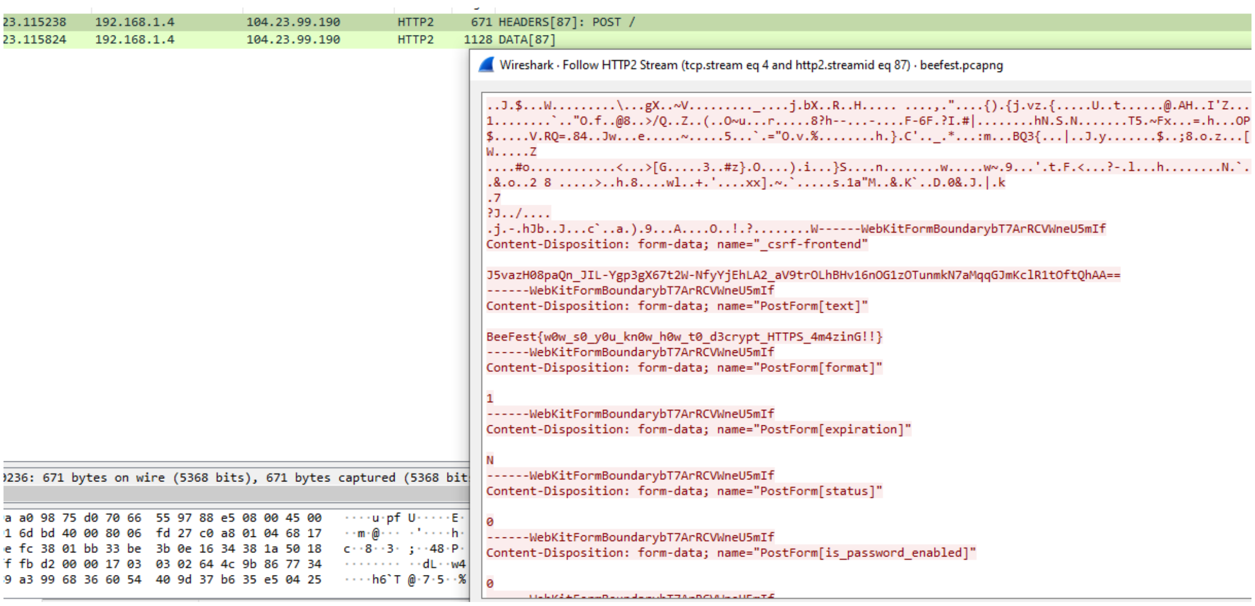


```
    Time        Source          Destination      Protocol  Length Info
285 7.194865    192.168.1.4     104.23.99.190    HTTP2     146 Magic, SETTINGS[0], WINDOW_UPDATE[0]
286 7.195230    192.168.1.4     104.23.99.190    HTTP2     544 HEADERS[1]: GET /
290 7.216090    192.168.1.4     104.23.99.190    HTTP2      85 SETTINGS[0]
306 7.642171    192.168.1.4     104.23.99.190    HTTP2     278 HEADERS[3]: GET /rob
343 7.851974    192.168.1.4     104.23.99.190    HTTP2     203 HEADERS[5]: GET /ass
344 7.853407    192.168.1.4     104.23.99.190    HTTP2     132 HEADERS[7]: GET /ass
347 7.861695    192.168.1.4     104.23.99.190    HTTP2     136 HEADERS[9]: GET /ass
349 7.866051    192.168.1.4     104.23.99.190    HTTP2     138 HEADERS[11]: GET /as                            s
355 7.868223    192.168.1.4     104.23.99.190    HTTP2     136 HEADERS[13]: GET /as
374 7.889847    192.168.1.4     104.23.99.190    HTTP2     150 HEADERS[15]: GET /th                            a0b6023b5e6c9982d
387 7.895841    192.168.1.4     104.23.99.190    HTTP2     148 HEADERS[17]: GET /th                            023b5e6c9982d
389 7.902750    192.168.1.4     104.23.99.190    HTTP2     135 HEADERS[19]: GET /th
392 7.909507    192.168.1.4     104.23.99.190    HTTP2     129 HEADERS[21]: GET /th
393 7.912263    192.168.1.4     104.23.99.190    HTTP2     134 HEADERS[23]: GET /as
405 7.919169    192.168.1.4     104.23.99.190    HTTP2     125 HEADERS[25]: GET /as
407 7.925481    192.168.1.4     104.23.99.190    HTTP2     135 HEADERS[27]: GET /as
408 7.928442    192.168.1.4     104.23.99.190    HTTP2     137 HEADERS[29]: GET /as
419 7.941160    192.168.1.4     104.23.99.190    HTTP2     135 HEADERS[31]: GET /as
420 7.942165    192.168.1.4     104.23.99.190    HTTP2     132 HEADERS[33]: GET /as
421 7.949540    192.168.1.4     104.23.99.190    HTTP2     150 HEADERS[35]: GET /th
432 7.951158    192.168.1.4     104.23.99.190    HTTP2     147 HEADERS[37]: GET /th
513 8.069758    192.168.1.4     104.26.14.238    HTTP2     146 Magic, SETTINGS[0], 
514 8.070164    192.168.1.4     104.26.14.238    HTTP2     488 HEADERS[1]: GET /adv
562 8.120910    192.168.1.4     104.26.14.238    HTTP2      85 SETTINGS[0]
730 8.501116    192.168.1.4     52.54.154.179    HTTP2     153 Magic, SETTINGS[0], WINDOW_UPDATE[0]
731 8.501421    192.168.1.4     52.54.154.179    HTTP2     582 HEADERS[1]: POST /log/wp/
```

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream

If you look at the next streams, there are indeed those that are still unreadable because wireshark may also capture some **ads** or **images** or **videos**. Those unreadable part are not encrypted data, but they are in **bytes format**.

But, since the stream hasn't finished yet, we can continue searching for anything that is interesting. And it turns out that there is a human readable HTTP body on the 87th stream.



```
23.115238    192.168.1.4     104.23.99.190     HTTP2     671 HEADERS[87]: POST /
23.115824    192.168.1.4     104.23.99.190     HTTP2    1128 DATA[87]
```

Wireshark · Follow HTTP2 Stream (tcp.stream eq 4 and http2.streamid eq 87) · beefest.pcapng

```
..J.$...W.........\...gX..~V........._....j.bX..R..H.....  ....,."....{).{j.vz.{.....U..t......@.AH..I'Z...
1........`.."O.f..@8..>/Q..Z..(..O~u...r.....8?h--...-....F-6F.?I.#|........hN.S.N.......T5.~Fx...=.h...OP
$.....V.RQ=.84..Jw...e.....~.....5...`.="O.v.%........h.}.C'.._.*...:m...BQ3{...|..J.y.......$..;8.o.z...[
W.....Z
....#o............<...>[G.....3..#z}.O....).i...}S....n........w.....w~.9...'.t.F.<...?-.l...h........N.`.
.&.o..2 8 .....>..h.8....wl..+.'....xx].~.`.....s.1a"M..&.K`..D.0&.J.|.k
.7
?J../....
.j.-.hJb..J...c`..a.).9...A....O..!.?........W------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="_csrf-frontend"

J5vazH08paQn_JIL-Ygp3gX67t2W-NfyYjEhLA2_aV9trOLhBHv16nOG1zOTunmkN7aMqqGJmKclR1tOftQhAA==
------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="PostForm[text]"

BeeFest{w0w_s0_y0u_kn0w_h0w_t0_d3crypt_HTTPS_4m4zinG!!}
------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="PostForm[format]"

1
------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="PostForm[expiration]"

N
------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="PostForm[status]"

0
------WebKitFormBoundarybT7ArRCVWneU5mIf
Content-Disposition: form-data; name="PostForm[is_password_enabled]"

0
```

```
)236: 671 bytes on wire (5368 bits), 671 bytes captured (5368 bit

a a0 98 75 d0 70 66  55 97 88 e5 08 00 45 00   ····u·pf U·····E·
1 6d bd 40 00 80 06  fd 27 c0 a8 01 04 68 17   ··m·@···  ·'····h·
e fc 38 01 bb 33 be  3b 0e 16 34 38 1a 50 18   c··8··3·  ;··48·P·
f fb d2 00 00 17 03  03 02 64 4c 9b 86 77 34   ········  ··dL··w4
9 a3 99 68 36 60 54  40 9d 37 b6 35 e5 04 25   ····h6`T @·7·5··%
```

If you look carefully, you will find the Flag.

Flag : BeeFest{w0w_s0_y0u_kn0w_h0w_t0_d3crypt_HTTPS_4m4zinG!!}

---------------------------- DONE ----------------------------