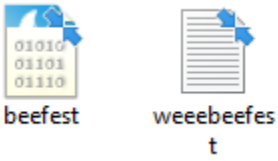
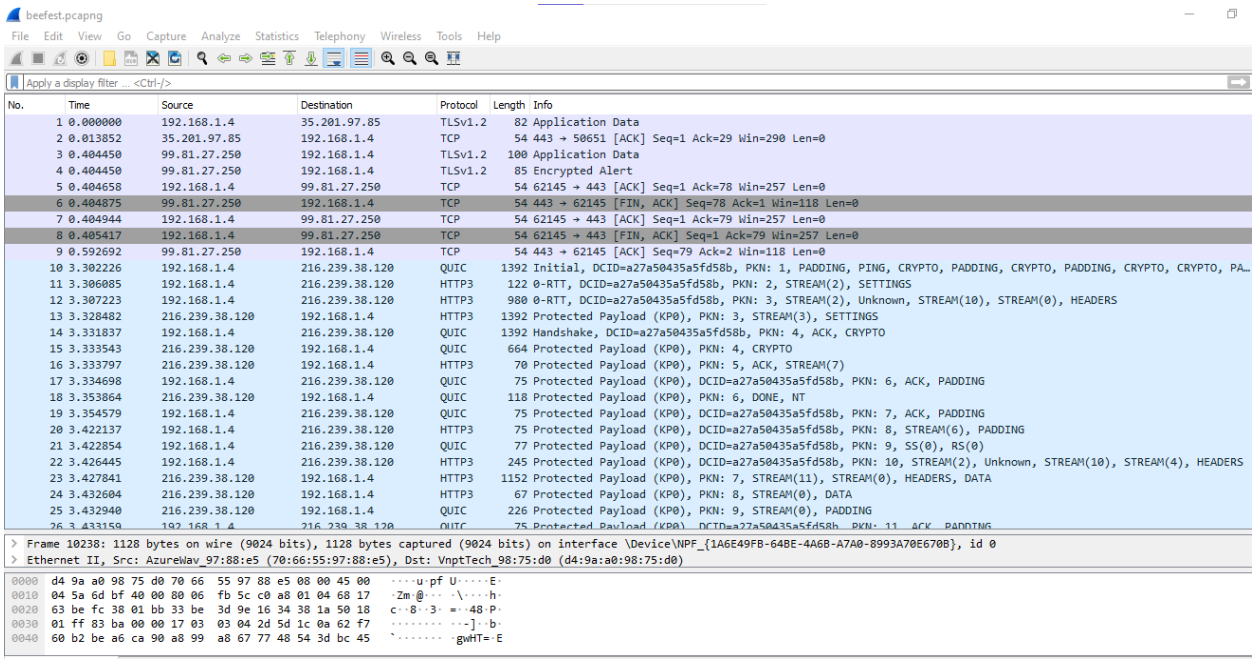
Too Sharky

Network Forensic Challenge

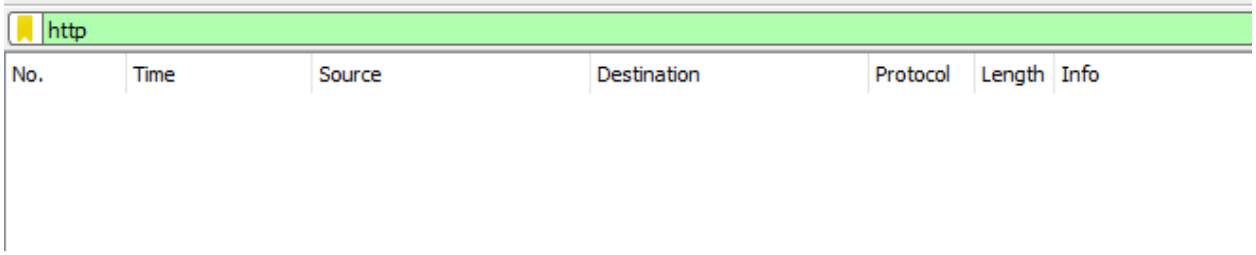
Pada chalenge ini, diberikan sebuah packet capture dan juga sebuah file text document.

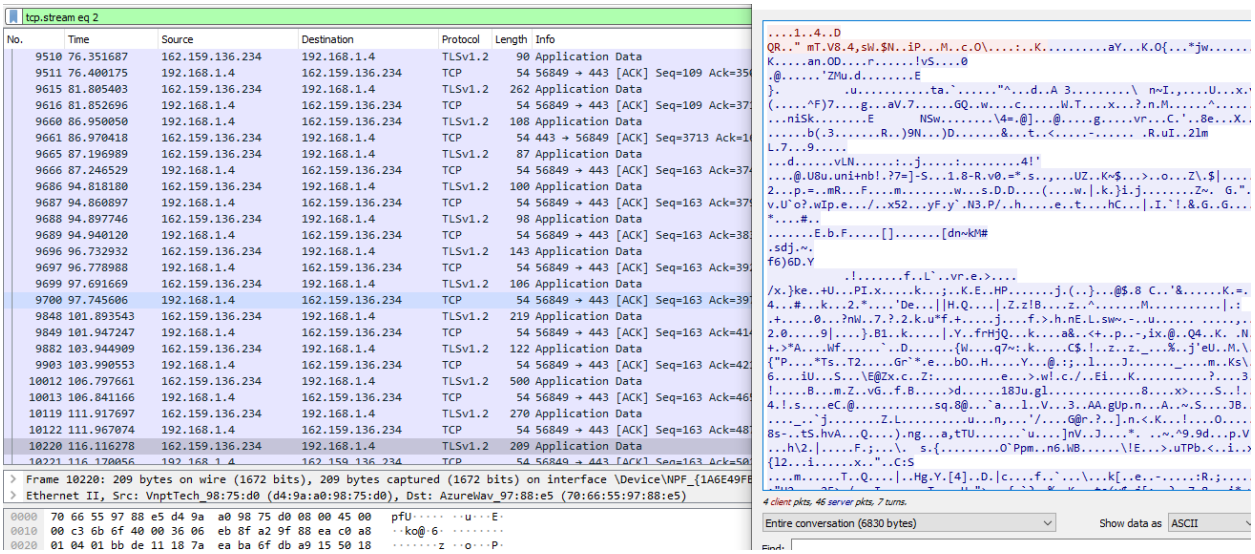


Mari kita analisa file packet capture nya dengan wireshark.

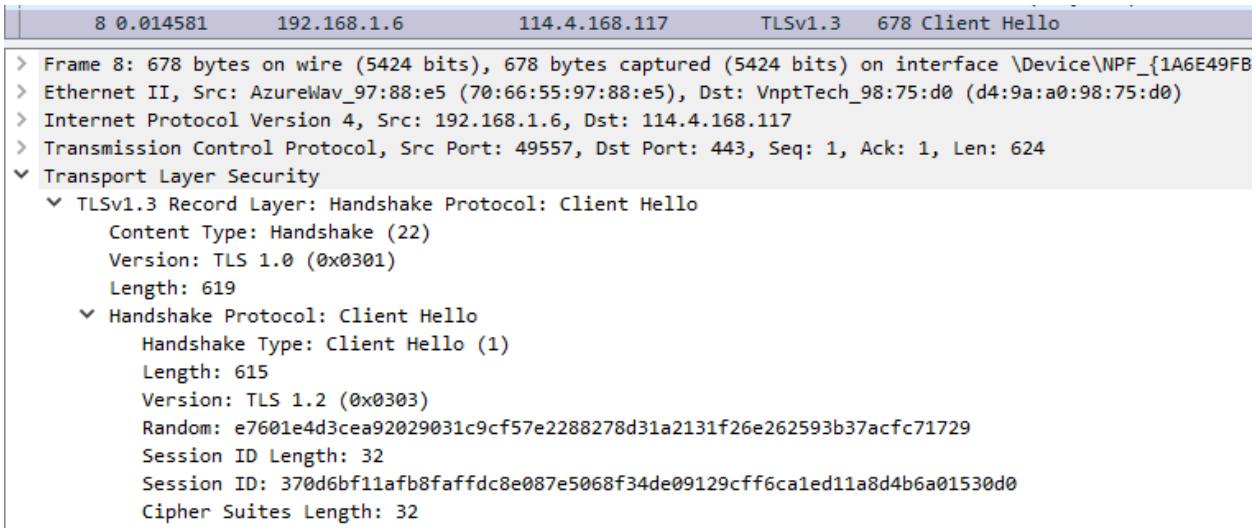


Setelah dianalisa sebentar, ternyata tidak ada protocol HTTP dan juga semua data dari client ke server dan sebaliknya tidak ada yang jelas.





Nah, bisa dilihat disini bahwa ada percakapan yang terenkripsi antara client dan server, yaitu client berbicara dengan server dengan protocol website HTTPS atau HTTP Secure dengan enkripsi TLSv1.3.



Juga percakapan yang terenkripsi tersebut datangnya dari IP Address 192.168.1.4

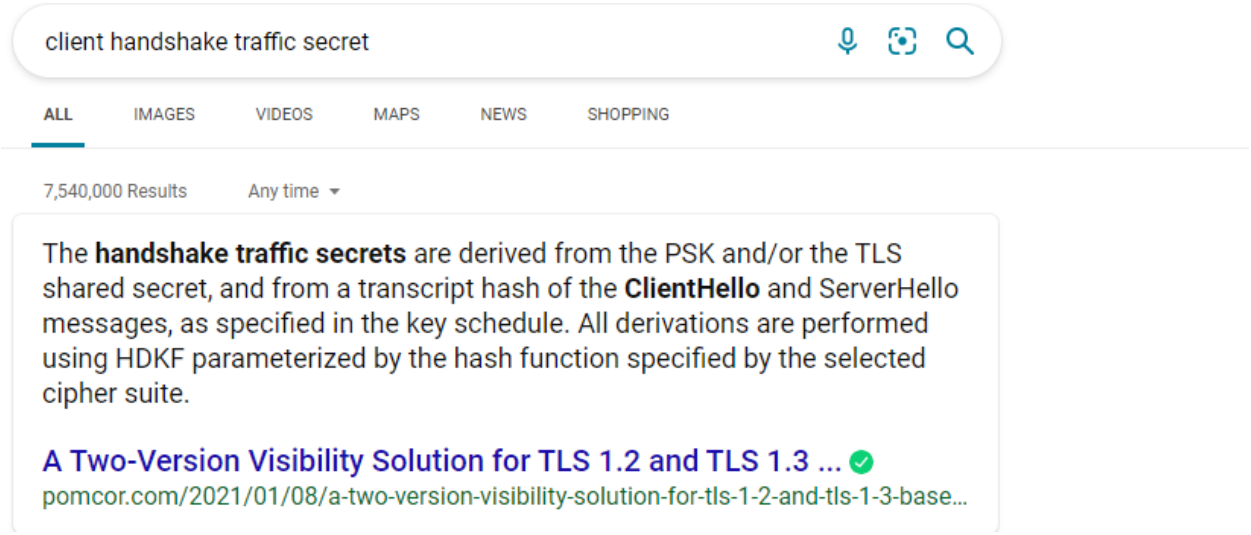
Gathered Information

Sejauh ini, informasi yang kita dapat adalah :
>> IP Address yang dicurigai = 192.168.1.4
>> Protocol Cryptographic / Enkripsi Traffic nya = TLSv1.3

Selanjutnya, bisa di cek ke file kedua, yang text document.
Disitu ada banyak sekali kata abstrak, kecuali kata-kata terdepan.

```
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 1998
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
```

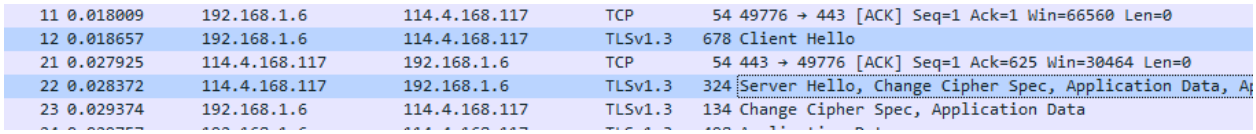
Jika bingung, bisa dicek di Google.
Contohnya seperti ini :



Setelah di search, kita juga bisa mengetahui suatu hal seperti **Client Hello** ataupun **Server Hello**

Dan mengapa IP Address yang dicurigai adalah 192.168.1.4?
Seperti yang sudah tertera, Client Hello berarti dari Client menyampaikan pesan atau request ke server atau sebuah endpoint, sedangkan Server Hello berarti mengirimkan pesan balik kepada Client.

Bisa dilihat pada gambar berikut :



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, A
23	0.029374	192.168.1.6	114.4.168.117	TLSv1.3	134 Change Cipher Spec, Application Data

IP Address sebelah kiri ada source, destination di sebelah kanan
...dengan Client Hello ada pada source yaitu 192.168.1.4

Kalau begini, yang harus kita lakukan adalah **mendecrypt traffic HTTPS.**

(Untuk referensi, bisa dilihat pada link berikut)
Wireshark Tutorial: Decrypting HTTPS Traffic (Includes SSL and TLS) (paloaltonetworks.com)

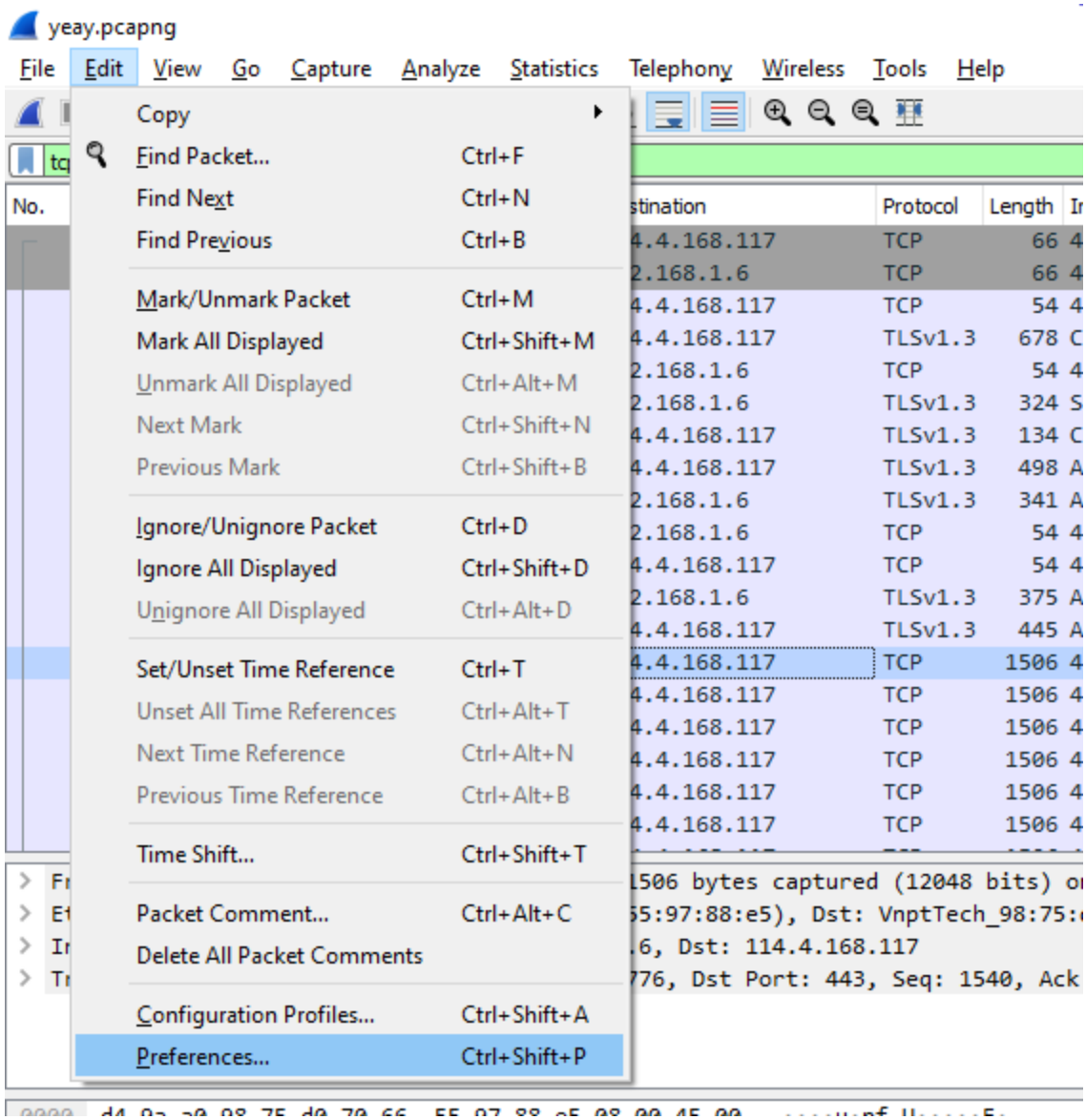
HTTPS Decryption

Dilihat dari link referensi sebelumnya, file document tersebut yang kita punya, berisikan SSL Key Log yang akan digunakan untuk mendecrypt TLS Encryptionnya. Tanpa adanya SSL Key Log terbuat selama proses packet capture dilakukan, maka traffic HTTPS yang berhasil dicapture pun akan menjadi sia-sia, kecuali hanya untuk melihat IP Address, Port, berapa banyak packet pada setiap send-data, dan sebagainya.

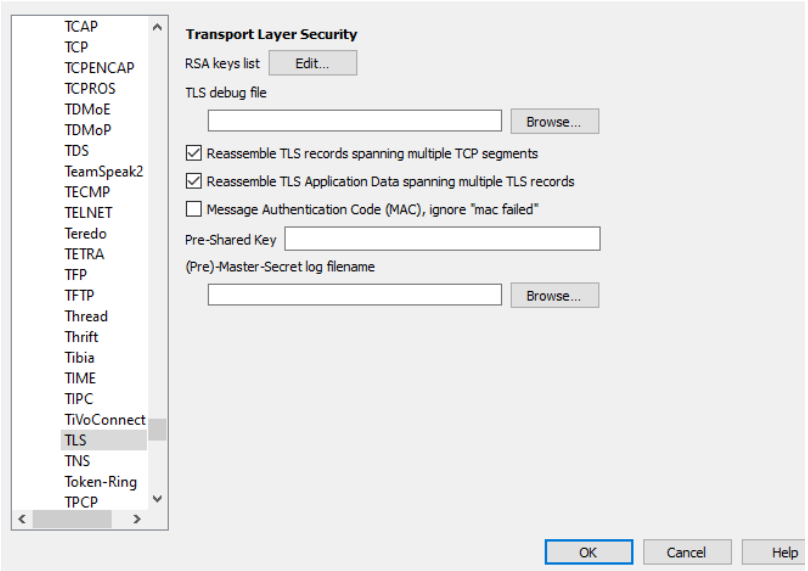
Karena yang terpenting adalah data-data apa saja yang bergerak dari Client ke Server dan sebaliknya yang terdapat pada header dan body dari website.

Berikut cara mendecrypt TLS dengan wireshark :

1. Ke Edit -> References



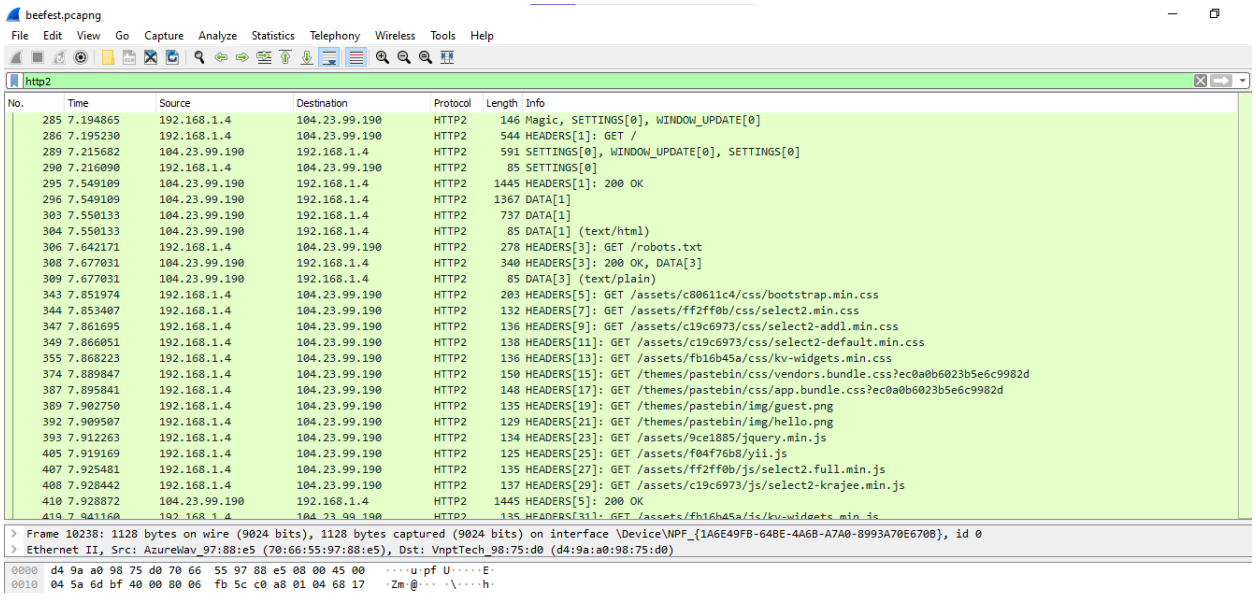
2. Selanjutnya, pada Protocols klik tombol drop-down dan pilih TLS



3. Kemudian klik browse dan masukkan SSL Key Log nya di bagian (Pre)-Master Secret log filename, yaitu file text berikut :

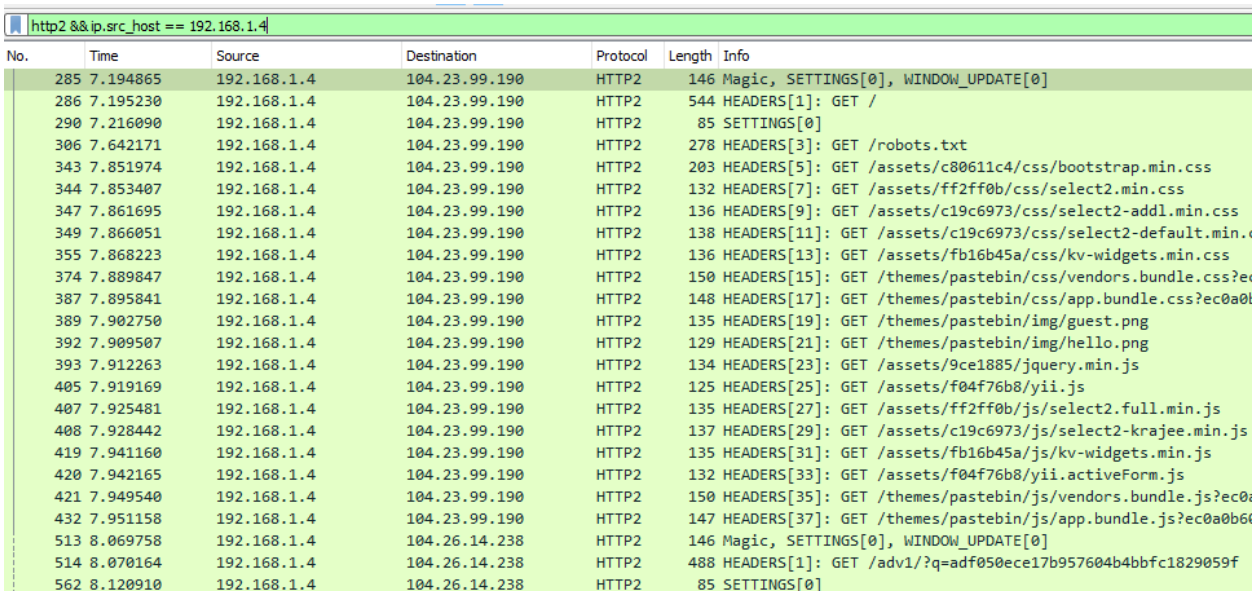
4. Setelah sudah, klik OK

Kemudian bisa di scroll-scroll dan sudah terlihat ada Traffic HTTP2 yang berarti HTTPS nya sudah berhasil di decrypt.

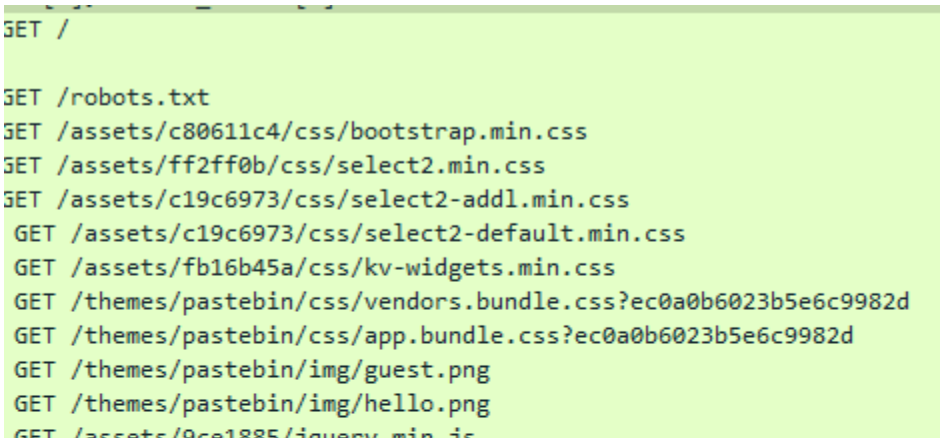


Lalu, bisa gunakan filter untuk mempercepat pencarian dengan syntax berikut :
>> http2 && ip.src_host == 192.168.1.4
Artinya, kita hanya akan menglist packet dengan protocol HTTP2 dan akan menempatkan IP Address 192.168.1.6 di kolom source.

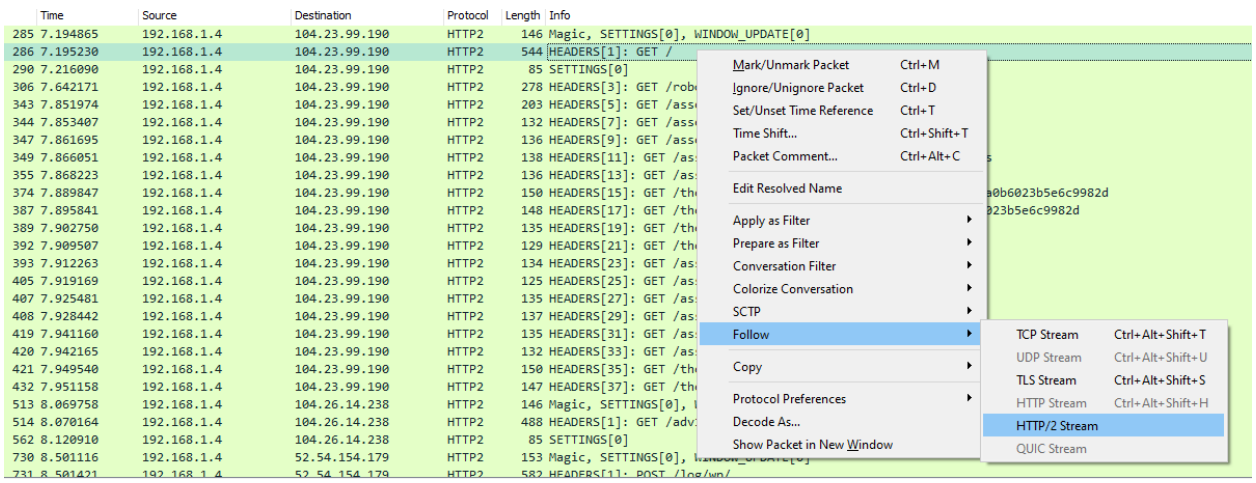
Berikut hasilnya :



Dari sini, bisa terlihat langsung bahwa IP 192.168.1.4 mengunjungi **website pastebin.com**

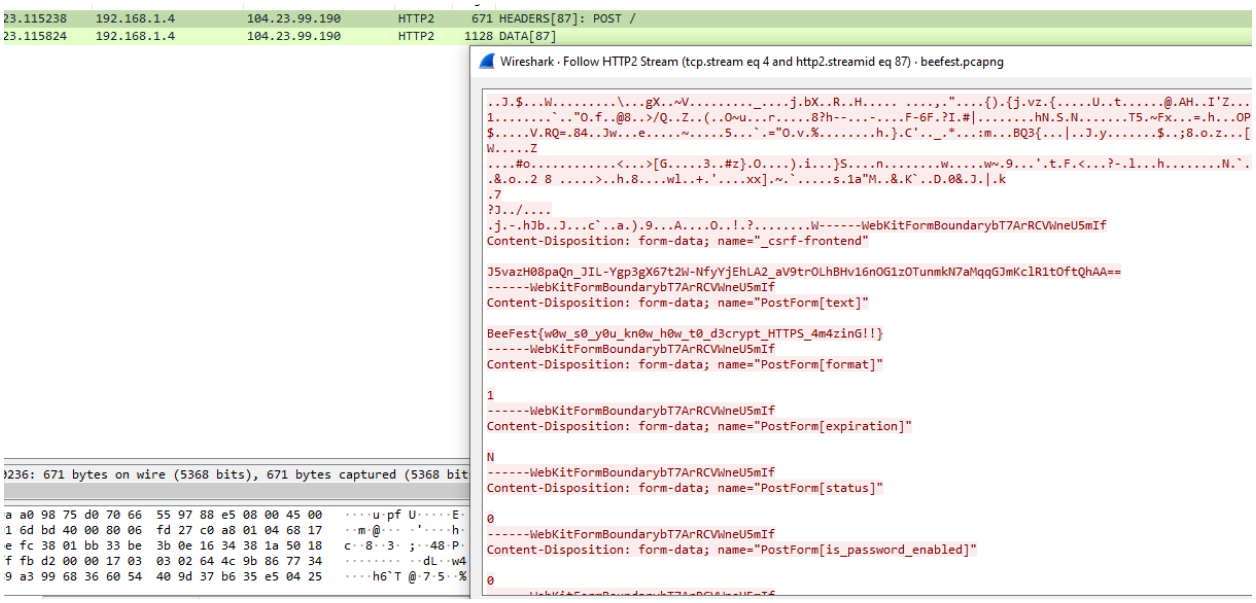


Sekarang coba kita follow HTTP/2 Stream nya



Jika dilihat ke stream-stream selanjutnya, memang ada yang masih terenkripsi karena wireshark mungkin juga mengcapture beberapa iklan atau gambar atau video yang memang tidak human-readable yang ter-load pada website.

Tapi, karena streamnya belum habis, kita bisa lanjutkan pencarian apakah ada sesuatu yang menarik. Dan ternyata ada body HTTP yang human readable pada stream ke 87.



Jika dilihat dengan teliti, maka didapatlah Flagnya.

Flag : BeeFest{w0w_s0_y0u_kn0w_h0w_t0_d3crypt_HTTPS_4m4zinG!!}

Challenge selesai

==========