



**Lembar Jawaban
Assignment - A02**

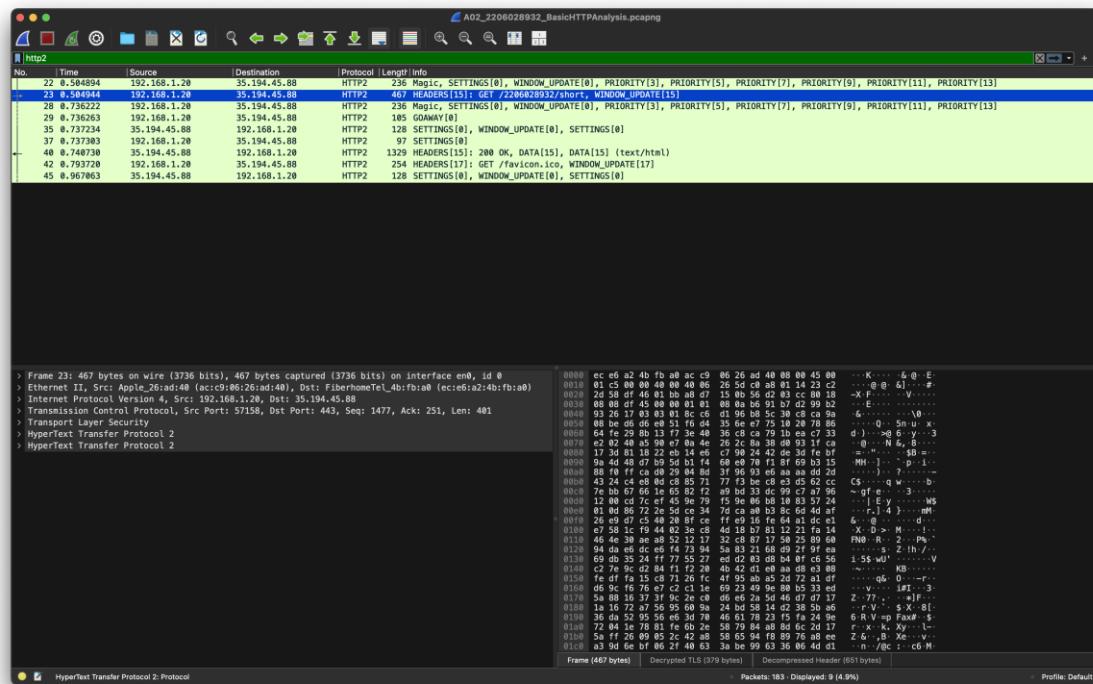
Webserver and HTTP Message Inspection

**Nama : Alden Luthfi
NPM : 2206028932**

[37 Points] Basic HTTP Analysis

1. [7] Temukan frame yang berisi HTTP request. Machine mana yang bertindak sebagai source dari pesan ini (apakah itu local machine atau web server Anda)? Jelaskan bagaimana Anda dapat menyimpulkannya dari informasi yang tersedia.

Screenshot:

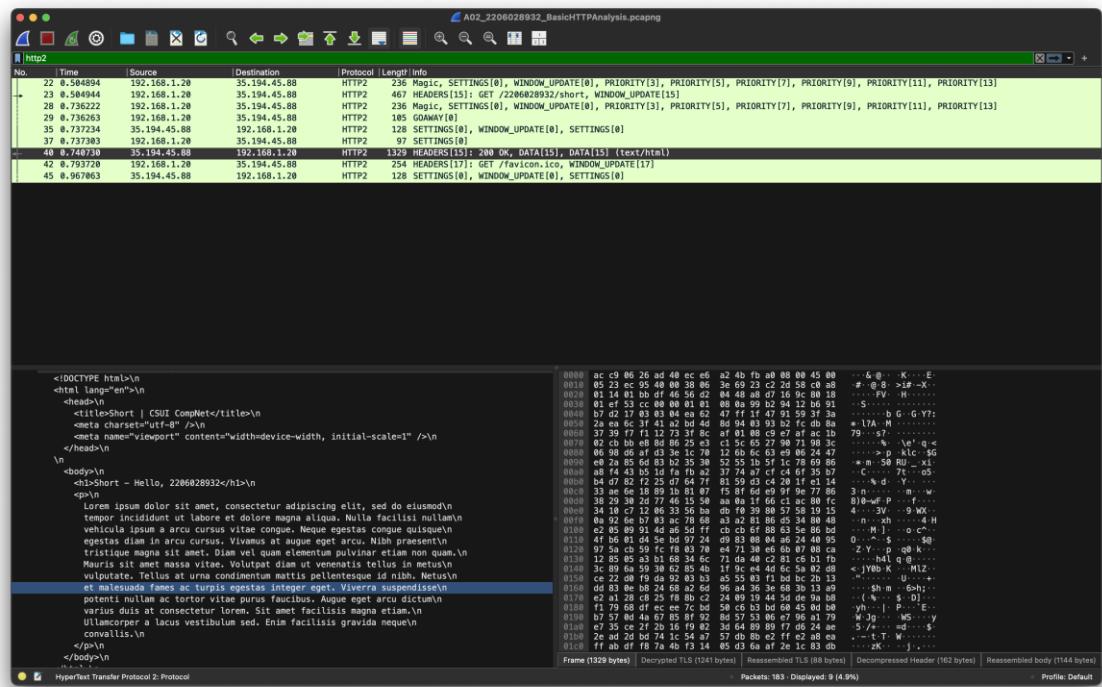
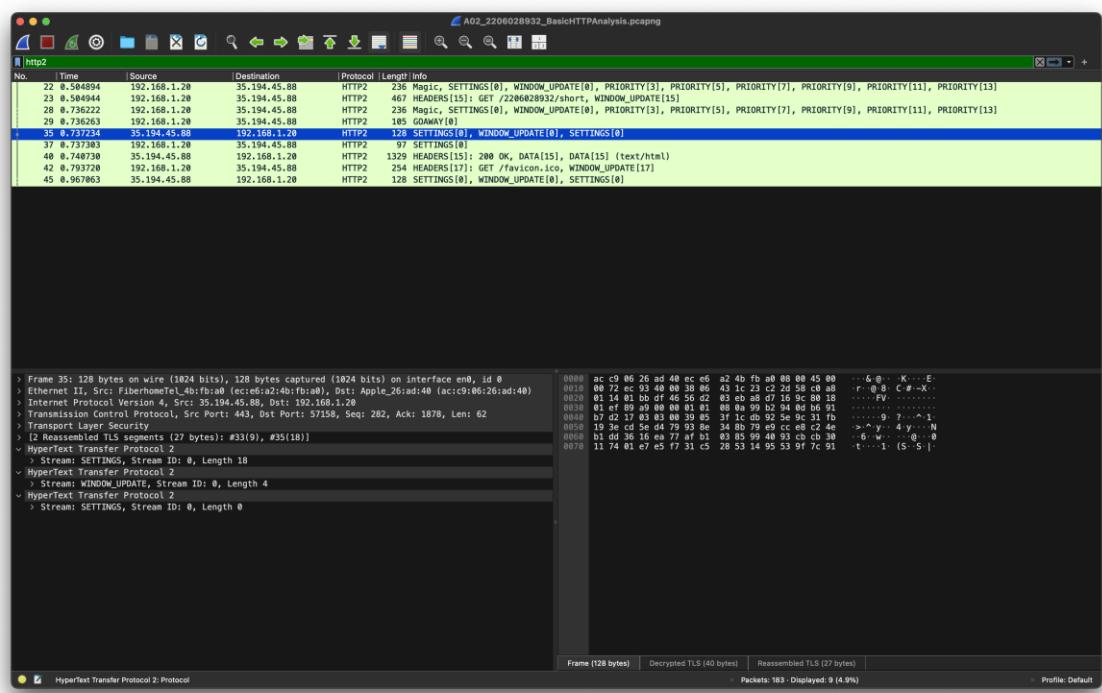


Explanation:

Baris dari *captured packets* yang di *highlight* biru (nomor 23) menunjukkan *request* HTTP GET dengan source IPv4 address mesin lokal 192.168.1.20. *Request* GET tersebut menuju ke alamat *web server* 35.194.45.88/2206028932/short. Semua informasi ini didapatkan dari kolom *source*, *destination*, dan *info* dari *captured packets*.

2. [6] Identifikasi dan laporan semua komponen pesan yang dikembalikan oleh server! Apakah seluruh pesan yang dikembalikan oleh server dimuat dalam satu paket?

Screenshot:



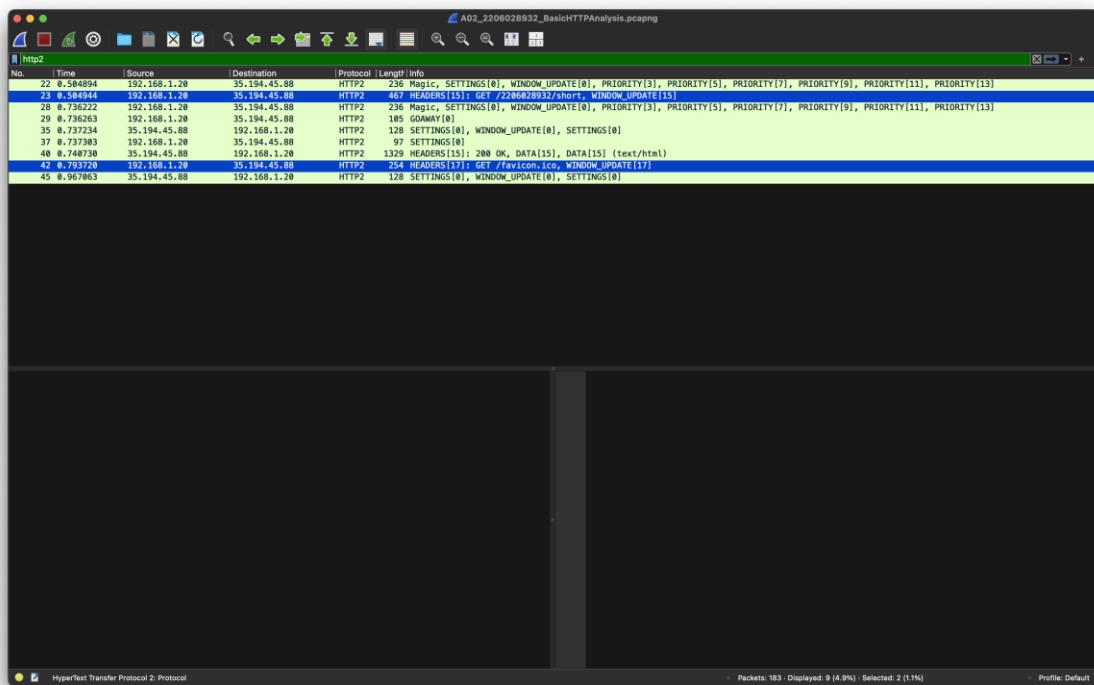
Explanation:

Contoh respons dari server terlihat pada packet dengan nomor 35, 40, dan 45. *Packet-packet* merupakan respons dari server dilihat dari *source*-nya yaitu 35.194.45.88 (IPv4 address VM). Pada packet 35 dan 45, SETTINGS[0] dan terutama WINDOW_UPDATE[0] menandakan status koneksi antara *client* dan *server*, biasanya *packet* ini muncul dari proses *refresh*. Sedangkan

packet 37 adalah packet yang sangat penting, *packet* tersebut berisi kode respons dari server (dalam kasus ini, server memberikan respons 200/OK yang berarti semua berjalan dengan lancar), selain itu, *packet* ini juga berisi data HTML yang akan ditampilkan di browser pengguna. Dari semua packets tersebut, bisa disimpulkan bahwa server tidak memberikan semua respons dalam satu packet saja.

- [6] Apa jenis HTTP request yang digunakan? Apakah request ini dianggap safe (tidak ada side effect)? Jelaskan alasannya!

Screenshot:



Explanation:

Kedua *request* yang dilakukan oleh user adalah *request GET*. Request GET HTTP tidak memiliki side effect, sehingga menurut definisi “aman” dari soal, *request GET* bisa dianggap “aman”. Namun, secara keseluruhan, *request GET* dianggap kurang aman daripada *request* lain seperti POST karena sifatnya yang mengambil suatu *resource* rawan terdampak serangan-serangan siber seperti *interception* dan *confused deputy attack*.

- [6] Bagaimana cara klien mengetahui fungsi dari bagian tertentu dalam pesan yang dikembalikan (misal sebagai header atau sebagai data)? Tandai bagian yang menunjukkan informasi tersebut!

Screenshot:

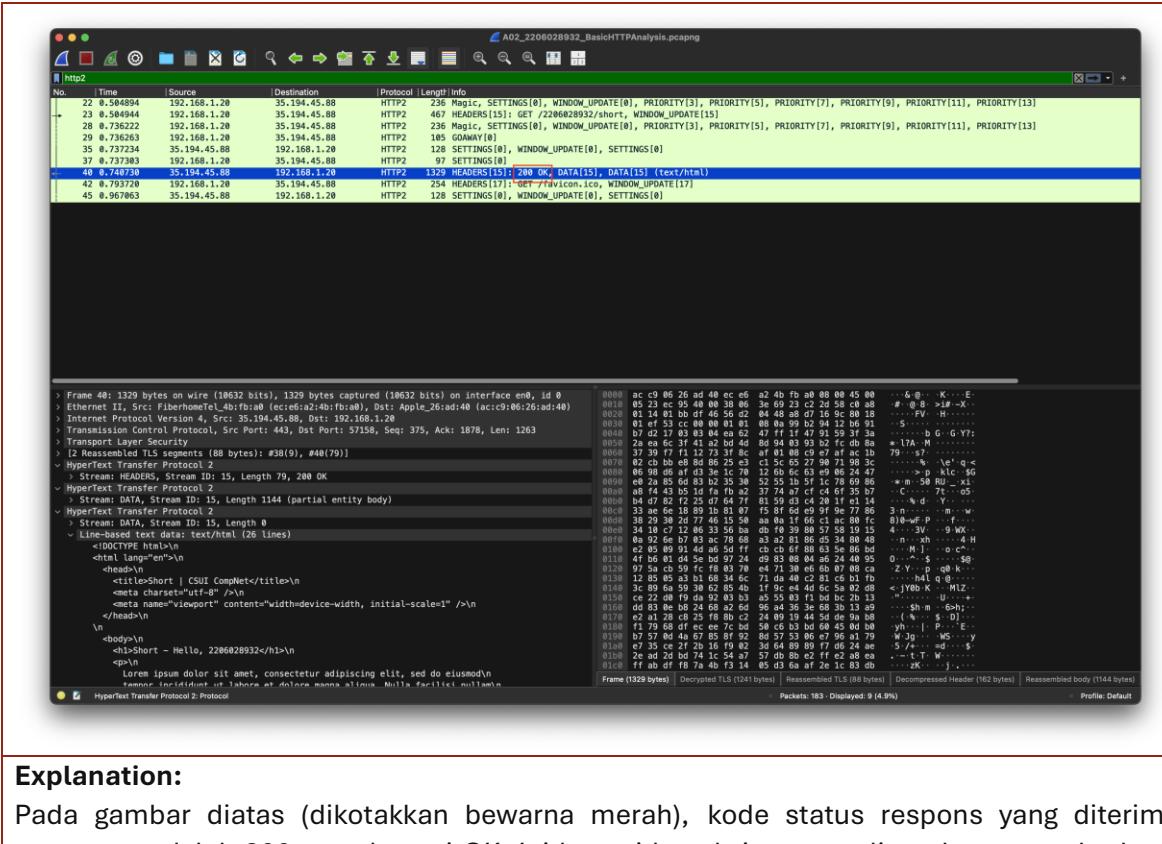
No.	Time	Source	Destination	Protocol	Length	Info
22	0.594894	192.168.1.28	35.194.45.88	HTTP2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
23	0.594944	192.168.1.28	35.194.45.88	HTTP2	467	HEADERS[15]: GET /228628932/short, WINDOW_UPDATE[15]
28	0.736222	192.168.1.28	35.194.45.88	HTTP2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
29	0.736263	192.168.1.28	35.194.45.88	HTTP2	185	GOAWAY[0]
35	0.737347	192.168.1.28	35.194.45.88	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
37	0.737383	192.168.1.28	35.194.45.88	HTTP2	97	SETTINGS[0]
48	0.740730	35.194.45.88	192.168.1.28	HTTP2	1329	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
42	0.793720	192.168.1.28	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
45	0.967063	35.194.45.88	192.168.1.28	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]

Explanation:

Pada kolom info, terdapat penanda-penanda yang memudahkan kita mengetahui peran dari masing-masing packet. Packet HEADER (dikotakkan berwarna biru) menandakan packet yang digunakan sebagai *header*. Ada juga packet HEADER yang ditandai dengan penanda DATA (dikotakkan berwarna pink), hal ini menandakan *packet* tersebut berisi data, dalam kasus ini, packet tersebut berisi data respons HTML yang dikirim ke pengguna.

- [6] Apa status code dari HTTP response dalam packet yang ditemukan? Apa arti dari status code dan phrase itu? Tandai bagian yang menunjukkan informasi tersebut!

Screenshot:

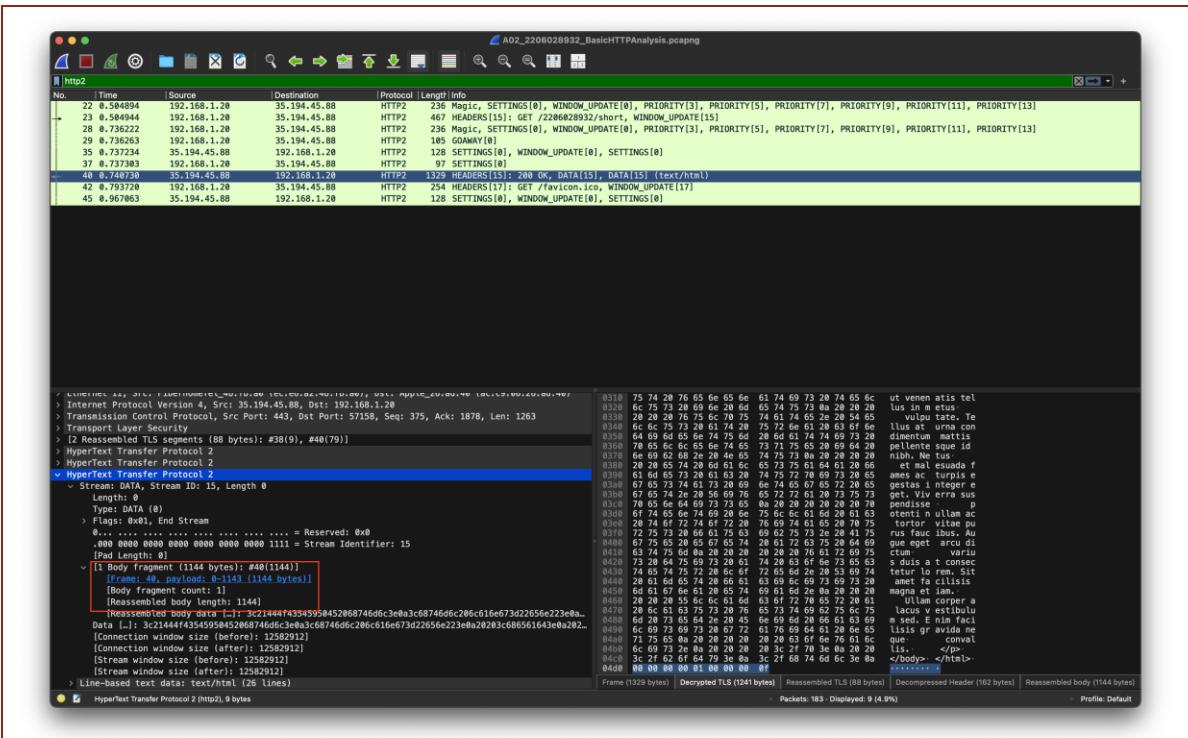


Explanation:

Pada gambar diatas (dikotakkan bewarna merah), kode status respons yang diterima pengguna adalah 200 yang berarti OK. Ini berarti koneksi antara *client* dan *server* berhasil berjalan dengan lancar.

- [6] Berapa body size dari HTTP response ke request /short?

Screenshot:



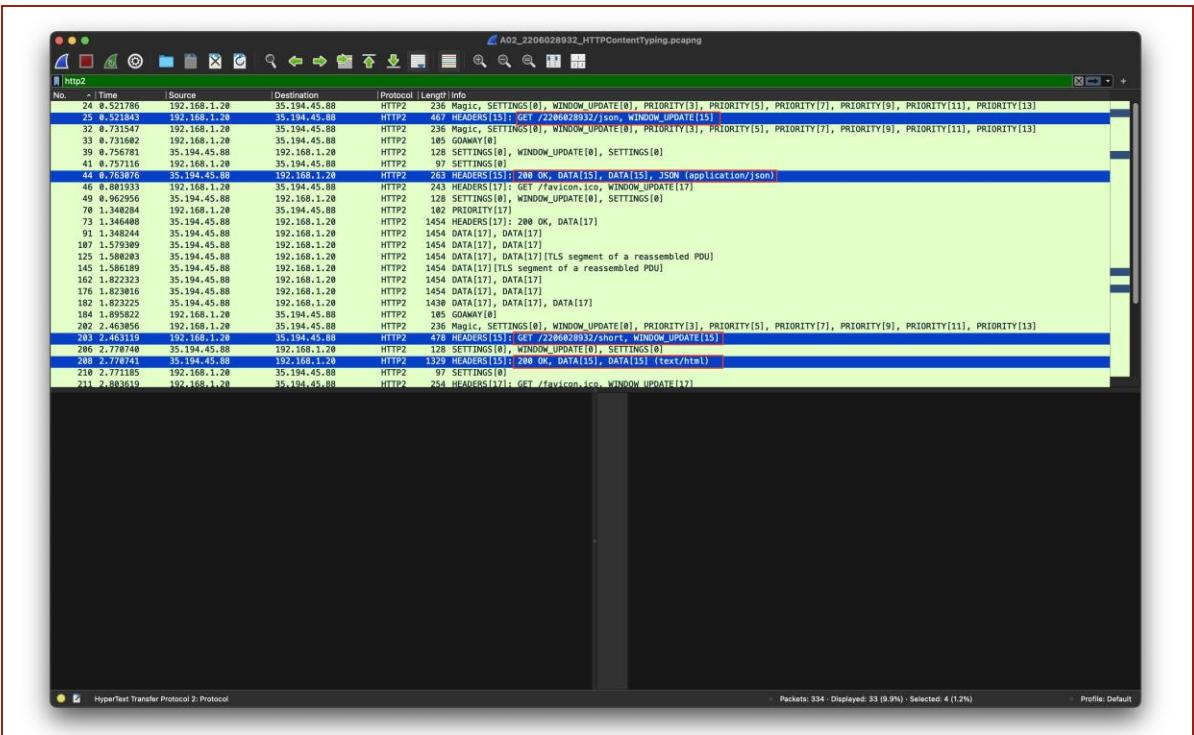
Explanation:

Berdasarkan kotak merah, ukuran packet yang diterima sebesar 1329 Bytes, sedangkan ukuran konten setelah TLS decryption (kanan bawah) adalah 1241 Bytes. Namun, jika dikulik lagi, terdapat bagian Body fragment dimana kita bisa menyimpulkan bahwa size dari response body request ke URL/short adalah 1144 Bytes.

[14 Points] HTTP Content Typing

1. [7] Bandingkan HTTP response dari dua request yang berbeda. Anda mungkin memperhatikan bahwa browser Anda dapat merender response secara berbeda. Bagaimana browser Anda mengetahui tentang content type yang diterimanya? (Petunjuk: periksa HTTP response)

Screenshot:



Explanation:

Dari respons yang diberikan server (packet yang dikotakkan merah bertuliskan 200 OK). Terdapat dua packet yang menampilkan konten yang berbeda, respons pertama mengirimkan “application/json” sedangkan respons kedua mengirimkan “text/html”.

2. [7] Apa content type dari URL /short? Bagaimana dengan content type dari URL /json?

Screenshot:

No.	Time	Source	Destination	Protocol	Length	Info
24	0.521786	192.168.1.28	35.194.45.88	HTTP/2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
25	0.521843	192.168.1.28	35.194.45.88	HTTP/2	467	HEADERS[15]: GET /226688932/.json, WINDOW_UPDATE[0], SETTINGS[0]
32	0.731547	192.168.1.28	35.194.45.88	HTTP/2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
33	0.731602	192.168.1.28	35.194.45.88	HTTP/2	185	GOAWAY[0]
39	0.731627	192.168.1.28	35.194.45.88	HTTP/2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
41	0.757136	192.168.1.28	35.194.45.88	HTTP/2	97	SETTINGS[0]
44	0.763876	35.194.45.88	192.168.1.28	HTTP/2	263	HEADERS[15]: 200 OK, DATA[15], DATA[15], JSON (application/json)
46	0.881933	192.168.1.28	35.194.45.88	HTTP/2	243	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
49	0.962956	35.194.45.88	192.168.1.28	HTTP/2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
78	1.346484	192.168.1.28	35.194.45.88	HTTP/2	145	HEADERS[17]: 200 OK, DATA[17]
79	1.346498	35.194.45.88	192.168.1.28	HTTP/2	1454	HEADERS[17]: 200 OK, DATA[17]
91	1.348244	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17], DATA[17]
107	1.579398	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17], DATA[17]
125	1.588203	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17], DATA[17] [TLS segment of a reassembled PDU]
145	1.588180	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17] [TLS segment of a reassembled PDU]
152	1.823833	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17], DATA[17]
176	1.823816	35.194.45.88	192.168.1.28	HTTP/2	1454	DATA[17], DATA[17]
182	1.823225	35.194.45.88	192.168.1.28	HTTP/2	1438	DATA[17], DATA[17], DATA[17]
184	1.895822	192.168.1.28	35.194.45.88	HTTP/2	185	GOAWAY[0]
202	2.463955	192.168.1.28	35.194.45.88	HTTP/2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
203	2.463119	192.168.1.28	35.194.45.88	HTTP/2	478	HEADERS[15]: GET /226688932/.short, WINDOW_UPDATE[0], SETTINGS[0]
206	2.463133	192.168.1.28	35.194.45.88	HTTP/2	133	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
208	2.771741	35.194.45.88	192.168.1.28	HTTP/2	1329	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
210	2.771185	192.168.1.28	35.194.45.88	HTTP/2	97	SETTINGS[0]
211	2.780359	192.168.1.28	35.194.45.88	HTTP/2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]

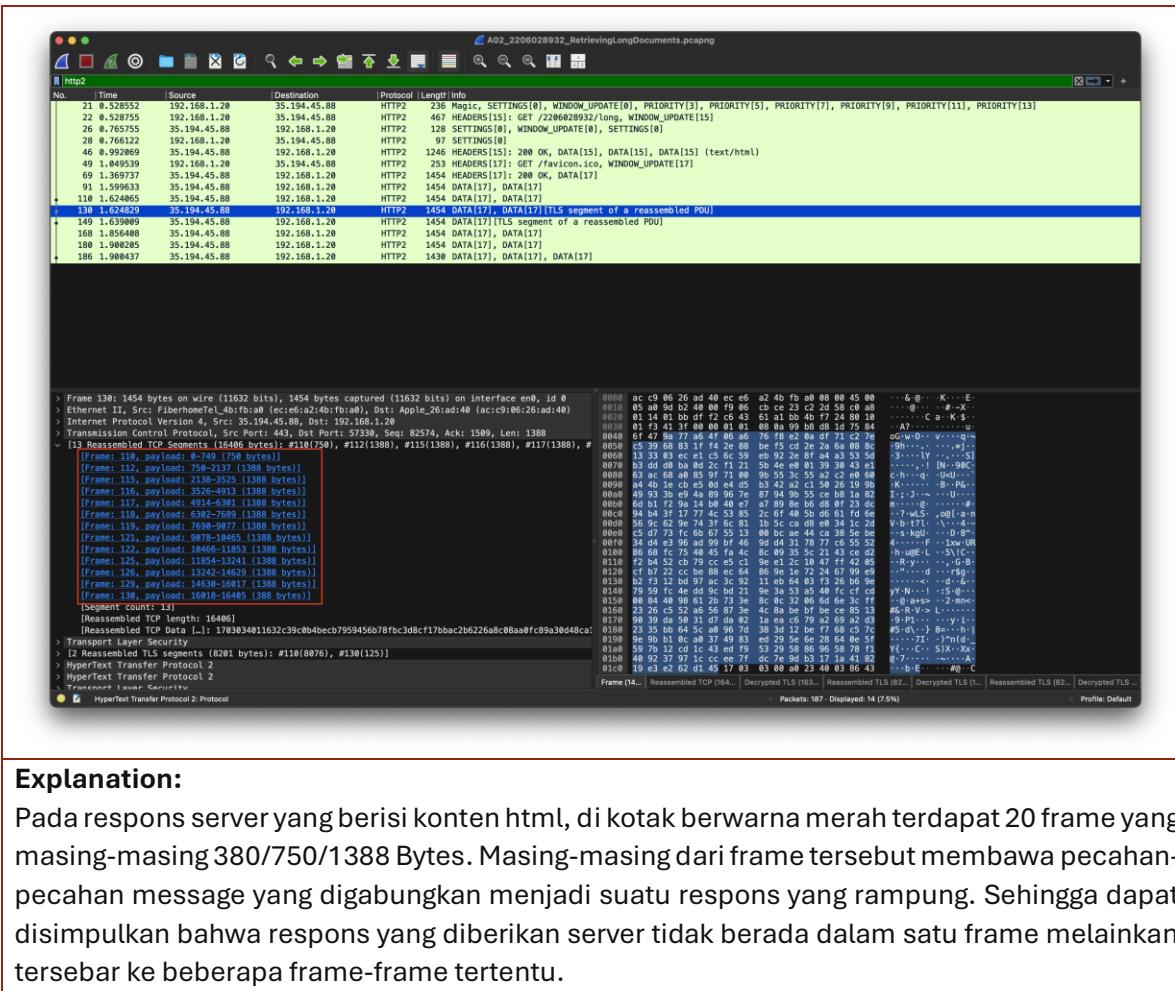
Explanation:

Content type dari URL/short adalah “text/html” sedangkan content type dari URL/json adalah “application/json”.

[14 Points] Retrieving Long Documents

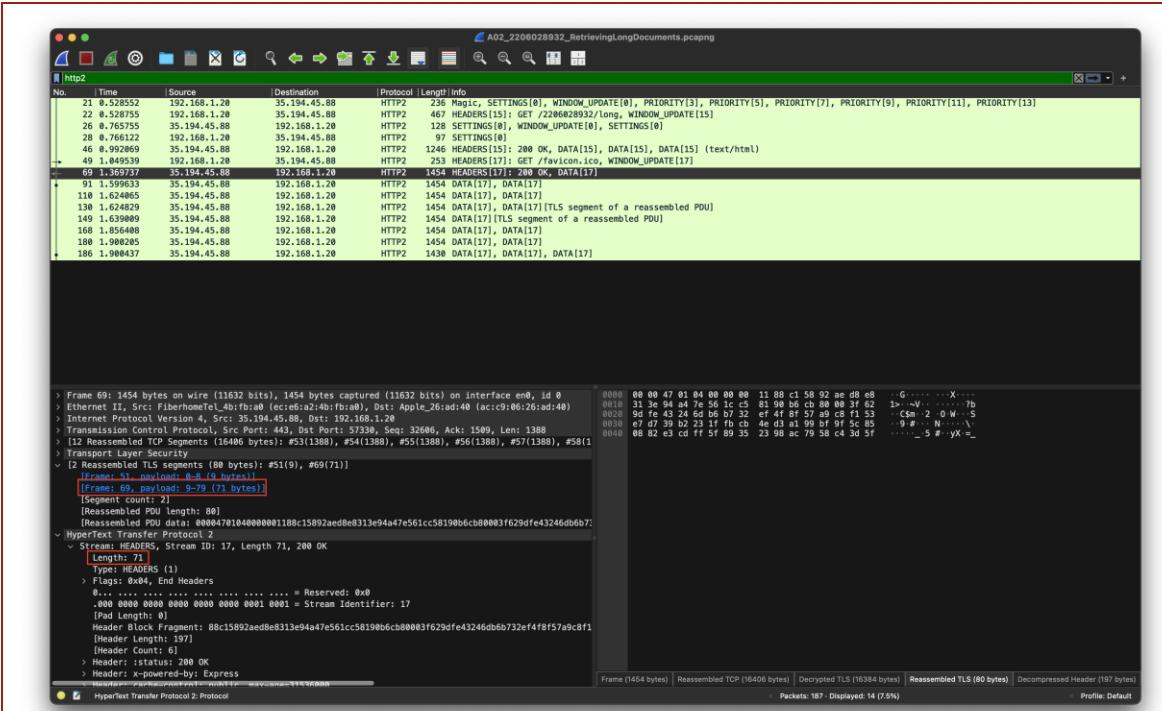
- [7] Dapatkah seluruh payload dari HTTP response message muat dalam satu frame? Jika ya, frame mana (sebutkan frame number) yang berisi keseluruhan response? Jika tidak, sebutkan semua frame number yang berpartisipasi dalam pengiriman HTTP response.

Screenshot:



- [7] Frame mana yang berisi HTTP response headers? (Anda mungkin perlu memeriksa contents dari frame)

Screenshot:



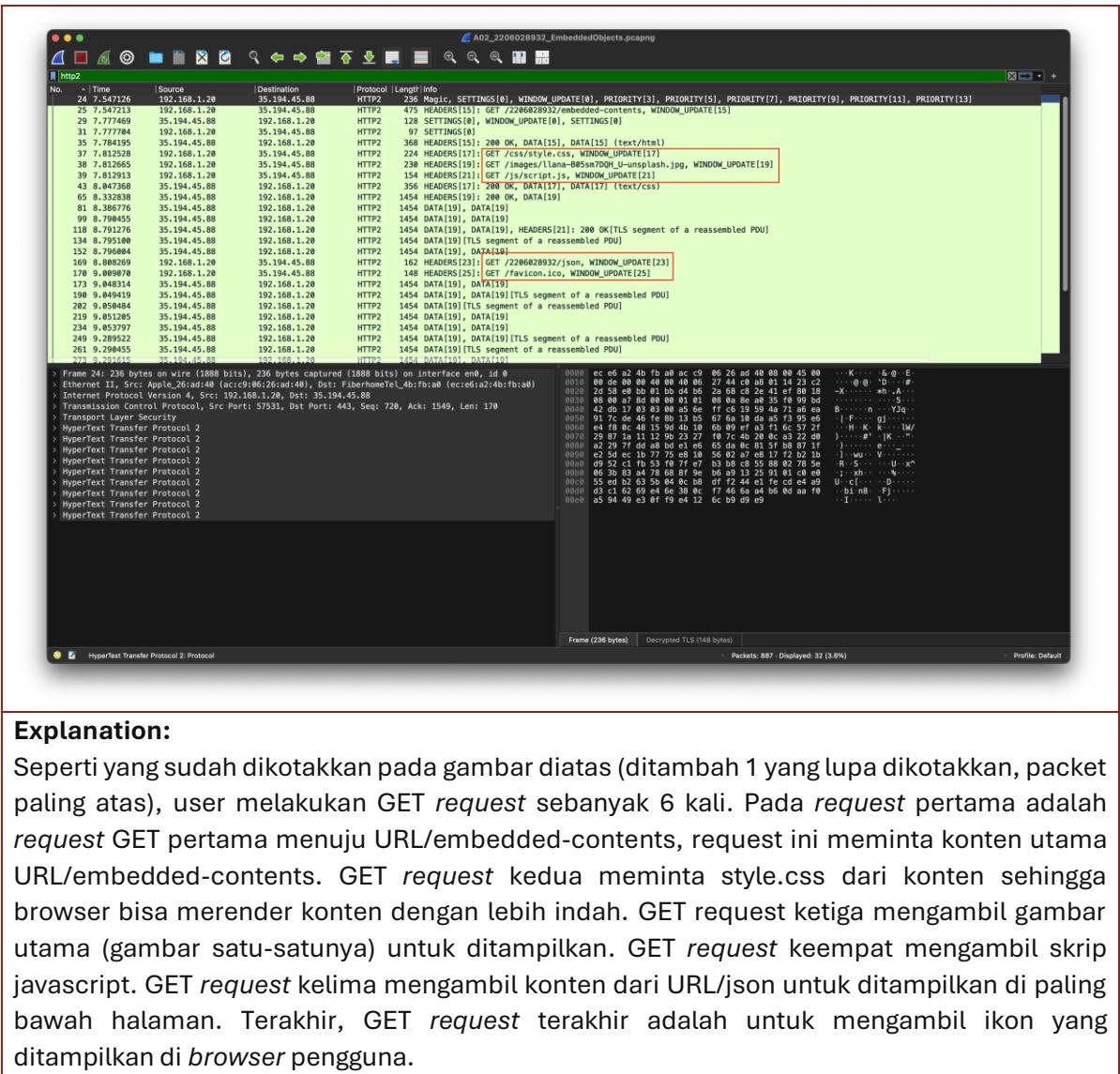
Explanation:

Seperti yang dikotakkan berwarna merah, terdapat 2 frame TLS yang direassemble yaitu frame 51 dan frame 69. Frame 69 mendapat perhatian lebih karena besarnya adalah 71 Bytes. Jika kita mengecek bagian header dari packet tersebut, bisa dicocokkan bahwa ukuran header juga 71 Bytes. Sehingga, bisa disimpulkan bahwa frame 69 berisi HTTP response header dari packet tersebut.

[14 Points] Content with Embedded Objects

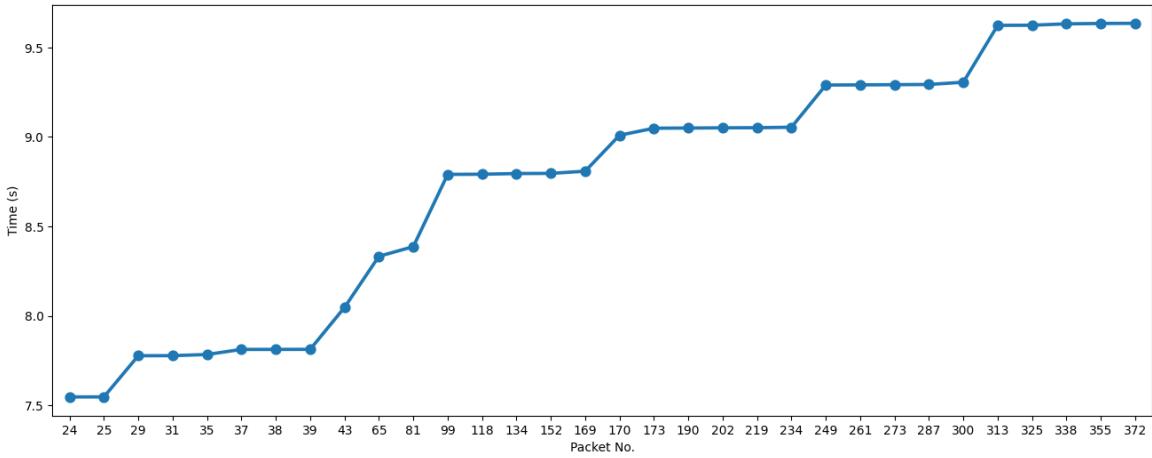
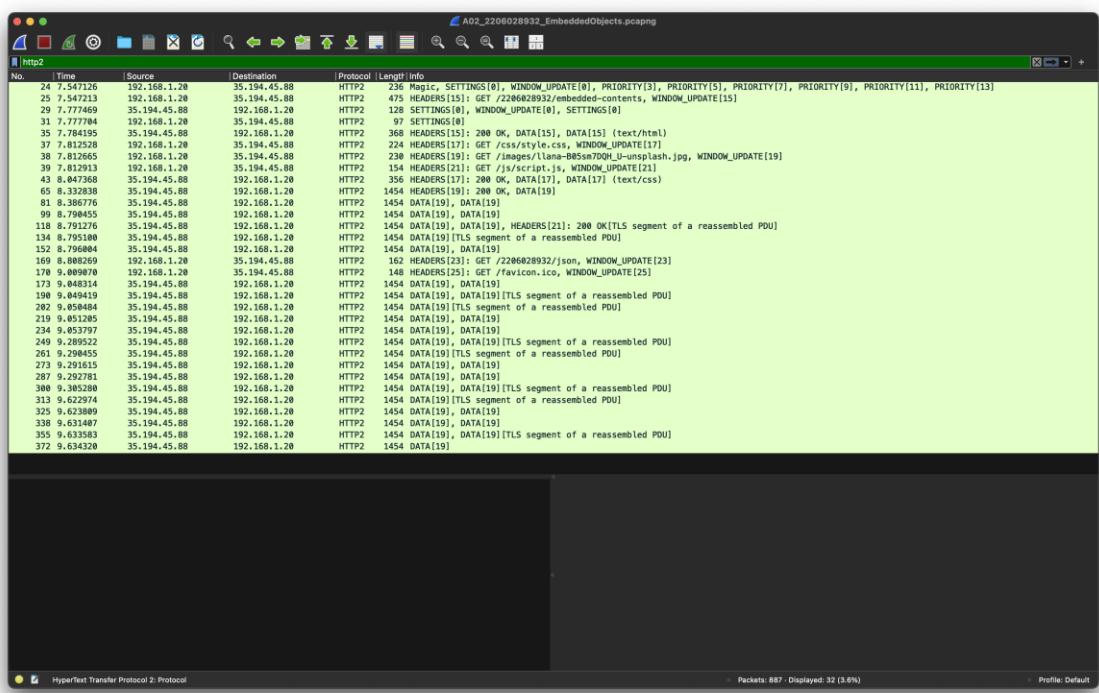
1. [7] Berapa banyak HTTP request yang dibuat dari source? Apa kegunaan setiap request (konten apa yang diminta dari setiap request)?

Screenshot:



2. [7] Perhatikan konten yang Anda unduh tersebut! Bisakah Anda menentukan apakah seluruh konten tersebut diambil secara bersamaan (secara paralel) atau serial, atau ada sebagian yang diunduh hanya setelah proses pengunduhan lainnya selesai? Jelaskan berdasarkan bukti yang didapatkan! [Petunjuk: Anda dapat membuktikannya dengan membuat grafik trafik berdasarkan data dari Wireshark, di mana sumbu Y menggambarkan objek request dan sumbu X menunjukkan waktu]

Screenshot:



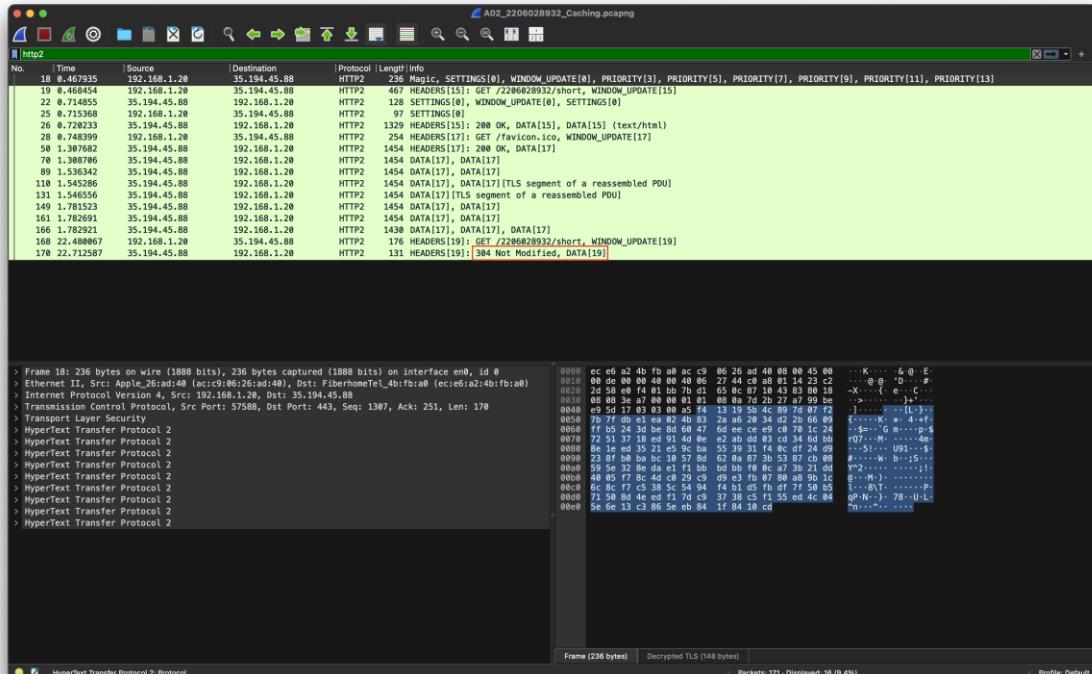
Explanation:

Pengiriman konten dilakukan secara pararel. Berdasarkan graf tersebut (paling terlihat di packet nomor 37-65) diketahui bahwa packet nomor 37, 38 dan 39 adalah GETrequest yang dilakukan *client*, namun, lonjakan waktu terjadi setelah ketiga request tersebut dijalankan. Jika pengiriman konten dilakukan secara serial, maka setiap melakukan *request*, *client* akan memakan waktu karena menunggu respons dari server. Namun, karena ketiga *request* terjadi sekaligus maka dapat disimpulkan *client* tidak menunggu *server* sebelum melakukan *request* lagi, sehingga pengiriman konten dilakukan secara pararel.

[21 Points] Persistent HTTP (Caching)

1. [5] Apa status code dan phrase dari HTTP response kedua? Apakah payload dikirim dari server dalam HTTP response tersebut? Jelaskan!

Screenshot:



Explanation:

Kode status resposn dari server setelah melakukan soft-refresh adalah *304 Not Modified*, ini artinya konten yang diminta oleh pengguna tidak berubah sejak terakhir diminta. Dalam kasus ini server tidak mengirim ulang konten tersebut, server juga memberi tahu *client* bahwa *cache* yang dimiliki *client* masih valid sehingga *client* masih bisa menggunakan ulang konten tanpa mengunduh ulang konten.

2. [6] Ada beberapa cara client dan server dapat menyepakati proses caching. Field apa dalam HTTP request kedua yang akan dicek untuk memutuskan apakah server harus mengirim payload?

Screenshot:

No.	Time	Source	Destination	Protocol	Length	Info
18	0.467935	192.168.1.20	35.194.45.88	HTTP2	238	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468454	192.168.1.20	35.194.45.88	HTTP2	467	HEADERS[15]: GET /2206028932/short, WINDOW_UPDATE[0], SETTINGS[0]
22	0.714855	35.194.45.88	192.168.1.20	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
25	0.715368	192.168.1.20	35.194.45.88	HTTP2	97	SETTINGS[0]
26	0.715373	192.168.1.20	35.194.45.88	HTTP2	1329	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
28	0.748399	192.168.1.20	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
58	1.307682	35.194.45.88	192.168.1.20	HTTP2	1454	HEADERS[17], DATA[17]
70	1.308786	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
89	1.536342	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
109	1.545286	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17] (TLS segment of a reassembled PDU)
131	1.546556	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] (TLS segment of a reassembled PDU)
149	1.781523	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
161	1.782691	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
166	1.782921	35.194.45.88	192.168.1.20	HTTP2	1438	DATA[17], DATA[17], DATA[17]
168	22.480067	192.168.1.20	35.194.45.88	HTTP2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
178	22.712587	35.194.45.88	192.168.1.20	HTTP2	131	HEADERS[19]: 304 Not Modified, DATA[19]

No.	Time	Source	Destination	Protocol	Length	Info
18	0.467935	192.168.1.20	35.194.45.88	HTTP2	238	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468454	192.168.1.20	35.194.45.88	HTTP2	467	HEADERS[15]: GET /2206028932/short, WINDOW_UPDATE[0], SETTINGS[0]
22	0.714855	35.194.45.88	192.168.1.20	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
25	0.715368	192.168.1.20	35.194.45.88	HTTP2	97	SETTINGS[0]
26	0.728233	35.194.45.88	192.168.1.20	HTTP2	1329	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
28	0.728239	192.168.1.20	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
58	1.307682	35.194.45.88	192.168.1.20	HTTP2	1454	HEADERS[17], DATA[17]
70	1.308786	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
89	1.536342	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
118	1.545286	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17] (TLS segment of a reassembled PDU)
149	1.546556	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] (TLS segment of a reassembled PDU)
161	1.782691	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
166	1.782921	35.194.45.88	192.168.1.20	HTTP2	1438	DATA[17], DATA[17], DATA[17]
168	22.480067	192.168.1.20	35.194.45.88	HTTP2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
178	22.712587	35.194.45.88	192.168.1.20	HTTP2	131	HEADERS[19]: 304 Not Modified, DATA[19]

No.	Time	Source	Destination	Protocol	Length	Info
18	0.467935	192.168.1.20	35.194.45.88	HTTP2	238	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468454	192.168.1.20	35.194.45.88	HTTP2	467	HEADERS[15]: GET /2206028932/short, WINDOW_UPDATE[0], SETTINGS[0]
22	0.714855	35.194.45.88	192.168.1.20	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
25	0.715368	192.168.1.20	35.194.45.88	HTTP2	97	SETTINGS[0]
26	0.728233	35.194.45.88	192.168.1.20	HTTP2	1329	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
28	0.728239	192.168.1.20	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
58	1.307682	35.194.45.88	192.168.1.20	HTTP2	1454	HEADERS[17], DATA[17]
70	1.308786	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
89	1.536342	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
118	1.545286	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17] (TLS segment of a reassembled PDU)
149	1.546556	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] (TLS segment of a reassembled PDU)
161	1.782691	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
166	1.782921	35.194.45.88	192.168.1.20	HTTP2	1438	DATA[17], DATA[17], DATA[17]
168	22.480067	192.168.1.20	35.194.45.88	HTTP2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
178	22.712587	35.194.45.88	192.168.1.20	HTTP2	131	HEADERS[19]: 304 Not Modified, DATA[19]

Explanation:

Untuk menentukan apakah server harus mengirim ulang konten, terdapat metadata yaitu if-none-match yang menyocokkan suatu entity tag yang diminta user terhadap entity tag yang disimpan oleh server. Entity tag digenerasi dengan suatu hash function yang menandakan bahwa versi konten yang diminta oleh client masih sama dengan yang disimpan, dalam kasis

ini, server akan mengembalikan respons 304 Not Modified dan client tidak perlu mengunduh konten lagi.

3. [5] Dari mana value dari field HTTP request kedua yang telah Anda identifikasi di nomor 2 berasal? Secara spesifik, field mana yang menjadi acuan untuk mengisi field tersebut? Tandai bagian yang menunjukkan informasi tersebut!

Screenshot:

http2

No. | Time | Source | Destination | Protocol | Length | Info

18. 0.467530 192.168.1.20 35.194.45.88 HTTP2 236 Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19. 0.467544 192.168.1.20 35.194.45.88 HTTP2 47 SETTINGS[15]; GET /2286028932/favicon.ico, WINDOW_UPDATE[15]
20. 0.734855 35.194.45.88 192.168.1.20 HTTP2 128 SETTINGS[8], WINDOW_UPDATE[6], SETTINGS[0]
21. 0.734868 192.168.1.20 35.194.45.88 HTTP2 97 SETTINGS[8]
26. 0.728233 35.194.45.88 192.168.1.20 HTTP2 1329 HEADERS[15]: 200 OK, DATA[15], DATA[15] {text/html}
28. 0.748399 192.168.1.20 35.194.45.88 HTTP2 259 HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
50. 1.307682 35.194.45.88 192.168.1.20 HTTP2 1454 HEADERS[17]: 200 OK, DATA[17]
50. 1.307680 192.168.1.20 35.194.45.88 HTTP2 1454 HEADERS[17]: 200 OK, DATA[17]
59. 1.353642 35.194.45.88 192.168.1.20 HTTP2 1454 DATA[17], DATA[17]
118. 1.545286 35.194.45.88 192.168.1.20 HTTP2 1454 DATA[17], DATA[17] {TLS segment of a reassembled POU}
131. 1.546556 35.194.45.88 192.168.1.20 HTTP2 1454 DATA[17], DATA[17] {TLS segment of a reassembled POU}
141. 1.761523 35.194.45.88 192.168.1.20 HTTP2 1454 DATA[17], DATA[17]
161. 1.782691 35.194.45.88 192.168.1.20 HTTP2 1454 DATA[17], DATA[17]
166. 1.782921 35.194.45.88 192.168.1.20 HTTP2 1458 DATA[17], DATA[17], DATA[17]
168. 22.408867 192.168.1.20 35.194.45.88 HTTP2 176 HEADERS[19]: GET /2286028932/short, WINDOW_UPDATE[19]
170. 22.712587 35.194.45.88 192.168.1.20 HTTP2 131 HEADERS[19]: 304 Not Modified, DATA[19]

System: MC40253, Stream ID: 15, Length 79, 200 ms
Length: 79
Type: HEADERS [3]
> Flags: 0x84, End Headers
0... = Reserved: 0x0
.000 0000 0000 0000 0000 0000 1111 = Stream Identifier: 15
[Pad Length: 0]
Header Block Segment: 3fe11f884889f2b567f05b0b22d1fe86c1e6bb0a847f5f92497ca589d34d1f5a1271d882a60b
[Header Count: 162]
[Header Count: 6]
> Header table size update
> Header: iostat 0K
> Header: x-powered-by: Express
> Header: content-type: text/html; charset=utf-8
> Header: content-length: 1144
> Header: "Content-Type: text/html; charset=UTF-8"
[Request in frame: 19]
HyperText Transfer Protocol 2
Stream: DATA, Stream ID: 15, Length 1144 (partial entity body)
Length: 1144
Type: DATA (8)
> Flags: 0x80
0... = Reserved: 0x0
.000 0000 0000 0000 0000 0000 1111 = Stream Identifier: 15
[Pad Length: 0]
Header (http2.header), 32 bytes

Frame (1529 bytes) Decrypted TLS (1241 bytes) Reassembled TLS (88 bytes) Decompressed Header (162 bytes) Reassembled body (1144 bytes)

Packets: 171 Disposed: 16 (0.4%) Profile: Default

http://22.0.0.200#8932_Caching.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
18	0.000-0.935	35.194.45.88	192.168.1.20	HTTP/2	467	SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468458	192.168.1.20	35.194.45.88	HTTP/2	467	HEADERS[15]: GET /2206028932/short, WINDOW_UPDATE[15]
22	0.714855	35.194.45.88	192.168.1.20	HTTP/2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[8]
25	0.715368	192.168.1.20	35.194.45.88	HTTP/2	97	SETTINGS[8]
26	0.728233	35.194.45.88	192.168.1.20	HTTP/2	132	HEADERS[15]: 200 OK, DATA[15], DATA[15] (text/html)
27	0.730099	192.168.1.20	35.194.45.88	HTTP/2	75	DATA[17]: /favicon.ico, WINDOW_UPDATE[17]
50	1.397682	35.194.45.88	192.168.1.20	HTTP/2	1454	HEADERS[17]: 200 OK, DATA[17]
70	1.398786	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17]
89	1.395642	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17]
110	1.545286	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17] (TLS segment of a reassembled PDU)
131	1.544656	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17] (TLS segment of a reassembled PDU)
149	1.729223	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17]
161	1.728901	35.194.45.88	192.168.1.20	HTTP/2	1454	DATA[17], DATA[17]
166	1.728921	35.194.45.88	192.168.1.20	HTTP/2	1438	DATA[17], DATA[17], DATA[17]
168	22.480867	192.168.1.20	35.194.45.88	HTTP/2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
170	22.712587	35.194.45.88	192.168.1.20	HTTP/2	131	HEADERS[19]: 304 Not Modified, DATA[19]

Header: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36

Header: accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/pjpeg,image/svg+xml,application/xml;q=0.8,image/apng,*/*;q=0.5

Header: accept-encoding: gzip, deflate, br, zstd

Header: dnt: 1

Header: sec-gpc: 1

Header: upgrade-insecure-requests: 1

Header: user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36

Header: sec-fetch-mode: navigate

Header: sec-fetch-site: cross-site

Header: if-none-match: W/"478-YqTzRnAA4UgylVtBcNz1DNY"

Header: priority: u0, i

Header: te: trailers

[full request URI: https://35.194.45.88:2206#8932/short]

[Response in frame: 170]

HyperText Transfer Protocol 2

Stream: WINDOW_UPDATE, Stream ID: 19, Length 4

Length: 4

Type: WINDOW_UPDATE (8)

Flags: 0x00

0...: = Reserved: 0x0

.000 0000 0000 0000 0000 0001 001 = Stream Identifier: 19

0...: = Reserved: 0x0

400000 1000 0000 0000 0000 0000 0000 = Window Size Increment: 12451840

[Stream window size (before): 131072]

[Stream window size (after): 12582912]

Frame (176 bytes) Decrypted TLS (88 bytes) Decompressed Header (656 bytes)

Packets: 171 Discarded: 16 (0.4%)

Profile: Default

Header (http2 header) 32 bytes

Explanation:

Pada respons HTTP pertama setelah *hard-refresh*, server memberikan metadata *entity tag* (*etag*) pada respons tersebut. *Etag* inilah yang menjadi penanda apakah konten yang diberikan server terdapat modifikasi atau tidak. Pada request GET kedua, header akan mengandung metadata yang mengecek jika *etag* yang dikirim oleh client sama. Jika *etag* yang dikirim oleh

client menyamai etag yang server harapkan, maka server akan mengembalikan respons 304 not modified.

4. [5] Mengapa force reload dapat memaksa server untuk mengembalikan response? Apa yang membedakan HTTP request pertama dan kedua sehingga request pertama memiliki payload sedangkan yang kedua tidak? Jelaskan dan tandai bagian yang menunjukkan informasi tersebut!

Screenshot:

http2

No	Time	Source	Destination	Protocol	Length	Info
18	0.467935	192.168.1.20	35.194.45.88	HTTP2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468454	192.168.1.20	35.194.45.88	HTTP2	467	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[0]
22	0.714855	35.194.45.88	192.168.1.20	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
25	0.715368	192.168.1.20	35.194.45.88	HTTP2	97	SETTINGS[0]
26	0.715373	35.194.45.88	192.168.1.20	HTTP2	1329	HEADERS[19]: 200 OK, DATA[15], DATA[15] (text/html)
28	0.748399	192.168.1.20	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
50	1.307682	35.194.45.88	192.168.1.20	HTTP2	1454	HEADERS[17]: DATA[17]
70	1.308786	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
89	1.536342	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
104	1.545286	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] [TLS segment of a reassembled PDU]
131	1.546556	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] [TLS segment of a reassembled PDU]
149	1.781523	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
161	1.782691	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
166	1.782921	35.194.45.88	192.168.1.20	HTTP2	1438	DATA[17], DATA[17], DATA[17]
168	22.480067	192.168.1.20	35.194.45.88	HTTP2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
170	22.712587	35.194.45.88	192.168.1.20	HTTP2	131	HEADERS[19]: 304 Not Modified, DATA[19]

```

Weight: 41
[Weight real: 42]
Header Block Fragment [...] : 82058c6042038013cfb2261139ec4f418965b5c2fb4b4daef3d877abed07f66a281bd0da
[Header Length: 455]
[Header Count: 18]
> Header: :method: GET
> Header: :path: /2206028932/short
> Header: :authority: 35.194.45.88
> Header: accept: */*
> Header: user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:130.0) Gecko/201001 Firefox/130.0
> Header: accept-encoding: gzip, deflate, br, zstd
> Header: dnt: 1
> Header: sec-gpc: 1
> Header: upgrade-insecure-requests: 1
> Header: sec-fetch-dest: document
> Header: sec-fetch-mode: navigate
> Header: sec-fetch-site: cross-site
> Header: priority: u=0, i=0
> Header: cache-control: no-cache
> Header: Cache-Control: no-cache
> Header: te: trailers
> Full request URL: https://35.194.45.88/2206028932/short
> Response in frame: 26
* Monostate Transfer Protocol 2
Header (http2.header), 1 bytes

```

Frame (467 bytes) Decrypted TLS (379 bytes) Decompressed Header (651 bytes)

Packets: 171 - Displayed: 16 (0.4%) Profile: Default

http2

No	Time	Source	Destination	Protocol	Length	Info
18	0.467935	192.168.1.20	35.194.45.88	HTTP2	236	Magic, SETTINGS[0], WINDOW_UPDATE[0], PRIORITY[3], PRIORITY[5], PRIORITY[7], PRIORITY[9], PRIORITY[11], PRIORITY[13]
19	0.468454	192.168.1.20	35.194.45.88	HTTP2	467	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[0]
22	0.714855	35.194.45.88	192.168.1.20	HTTP2	128	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
25	0.715368	192.168.1.20	35.194.45.88	HTTP2	97	SETTINGS[0]
26	0.728233	35.194.45.88	192.168.1.20	HTTP2	1329	HEADERS[19]: 200 OK, DATA[15], DATA[15] (text/html)
28	0.748399	192.168.1.20	35.194.45.88	HTTP2	254	HEADERS[17]: GET /favicon.ico, WINDOW_UPDATE[17]
50	1.307682	35.194.45.88	192.168.1.20	HTTP2	1454	HEADERS[17]: DATA[17]
70	1.308786	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
89	1.536342	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
118	1.545286	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] [TLS segment of a reassembled PDU]
131	1.546556	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17] [TLS segment of a reassembled PDU]
149	1.781523	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
161	1.782691	35.194.45.88	192.168.1.20	HTTP2	1454	DATA[17], DATA[17]
166	1.782921	35.194.45.88	192.168.1.20	HTTP2	1438	DATA[17], DATA[17], DATA[17]
168	22.480067	192.168.1.20	35.194.45.88	HTTP2	176	HEADERS[19]: GET /2206028932/short, WINDOW_UPDATE[19]
170	22.712587	35.194.45.88	192.168.1.20	HTTP2	131	HEADERS[19]: 304 Not Modified, DATA[19]

```

Weight: 41
[Weight real: 42]
Header Block Fragment [...] : 82058c6042038013cfb2261139ec4f418965b5c2fb4b4daef3d877abed07f66a281bd0da
[Header Length: 456]
[Header Count: 17]
> Header: :method: GET
> Header: :path: /2206028932/short
> Header: :authority: 35.194.45.88
> Header: accept: */*
> Header: user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:130.0) Gecko/201001 Firefox/130.0
> Header: accept-encoding: gzip, deflate, br, zstd
> Header: dnt: 1
> Header: sec-gpc: 1
> Header: upgrade-insecure-requests: 1
> Header: sec-fetch-dest: document
> Header: sec-fetch-mode: navigate
> Header: sec-fetch-site: cross-site
> Header: if-none-match: W/478-qjZrhAAa4ugyBLVT8bCnznI2W#*
> Header: priority: u=0, i=0
> Header: te: trailers
> Full request URL: https://35.194.45.88/2206028932/short
> Response in frame: 26
* Monostate Transfer Protocol 2
Header (http2.header), 1 bytes

```

Frame (176 bytes) Decrypted TLS (88 bytes) Decompressed Header (656 bytes)

Packets: 171 - Displayed: 16 (0.4%) Profile: Default

Explanation:

Dari perbandingan diatas, pada request GET pertama, terdapat metadata header yaitu **cache-control: no-cache**, hal ini menandakan bahwa pengguna tidak akan menggunakan hasil simpanan sebelumnya dalam mengambil resource dari server. Sedangkan pada request GET

kedua, *metadata* tersebut berubah menjadi *if-none-match:[...]*, hal ini menandakan bahwa pengguna bisa saja menggunakan konten yang sama jika ada penanda yang sama.