

Laporan Umum
Workshop Administrasi Jaringan



Dosen Pengampu :
Dr. Ferry Astika Saputra, S.T., M.Sc.

Oleh:
Muhammad Rafi Dhiyaulhaq

312350004

4 D3 Teknik Informatika A

PROGRAM STUDI D3 TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

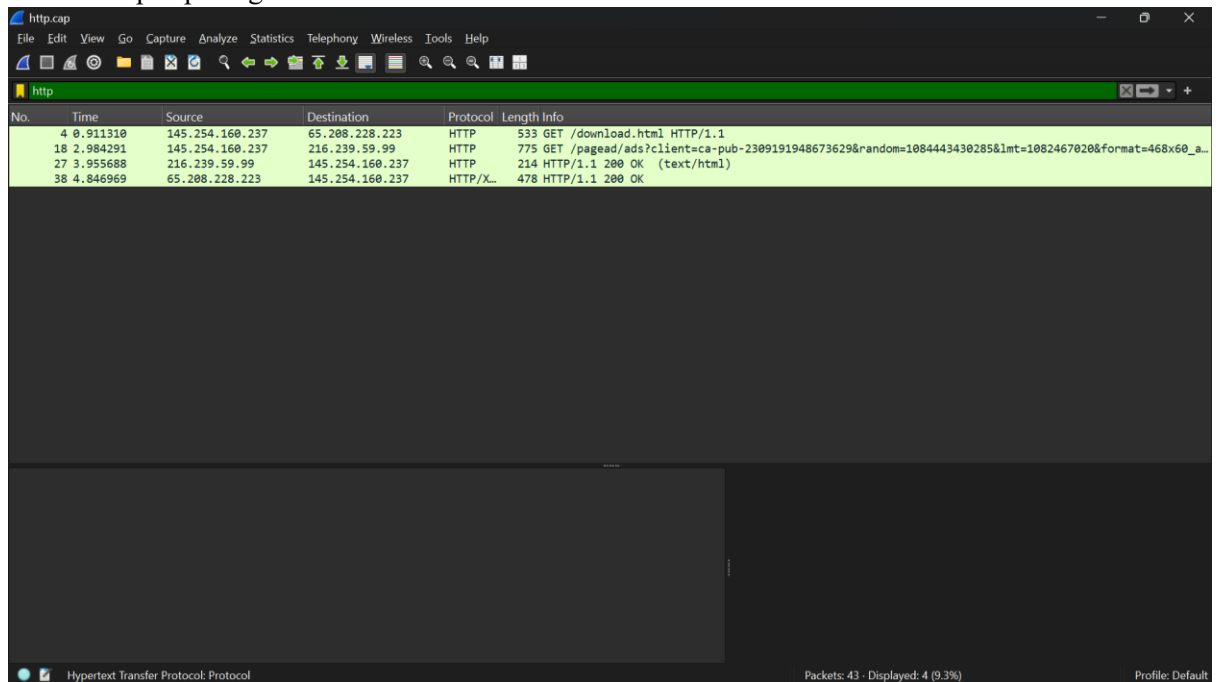
2024/2025

Tugas

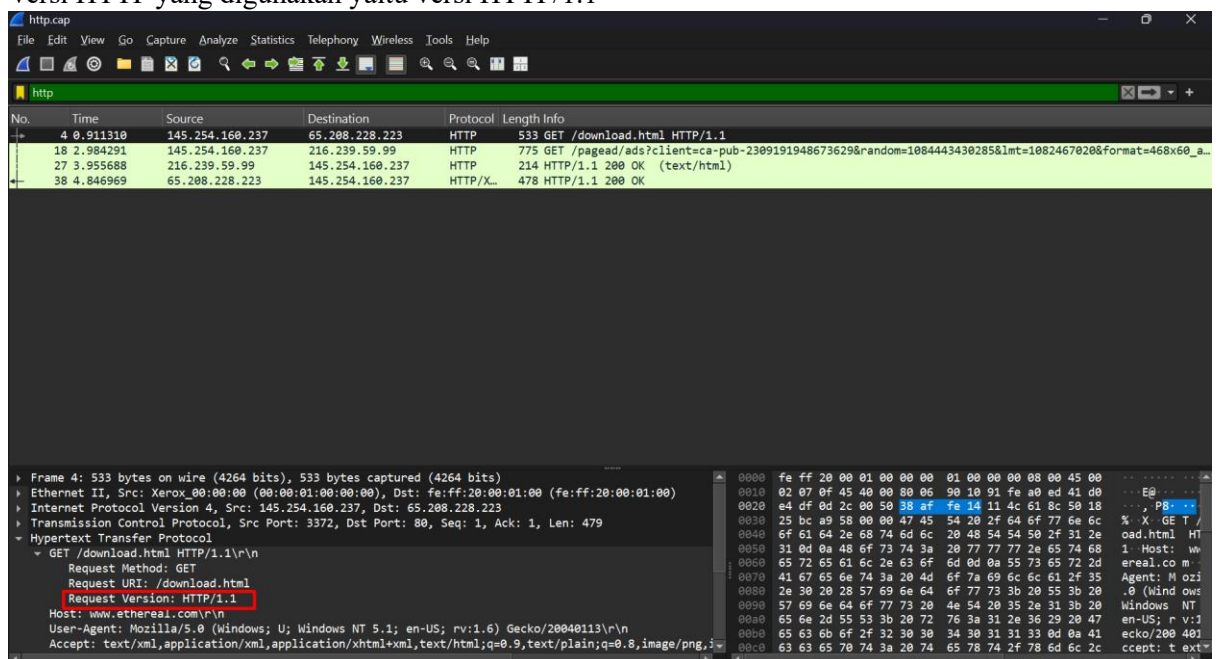
1. Analisa file http.cap dengan wireshark : Versi HTTP yang digunakan, IP address dari client maupun server, waktu dari client mengirimkan HTTP request., Waktu dari server mengirimkan server dan berapa durasinya
2. Deskripsi gambar pada slide
3. Rangkuman tahapan komunikasi menggunakan TCP

Jawaban

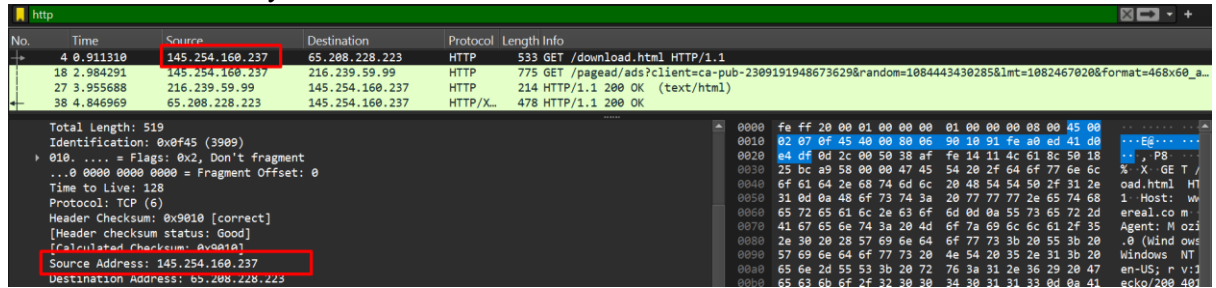
1. Analisa http.cap dengan wireshark



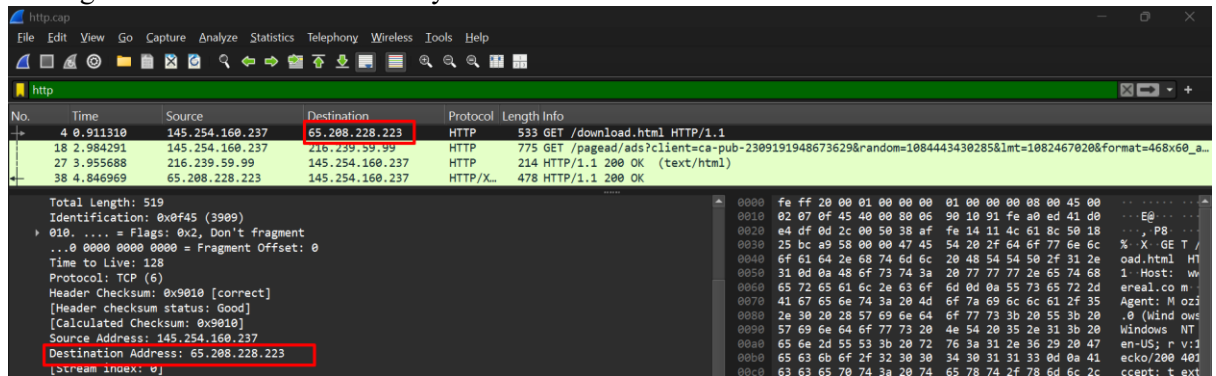
Versi HTTP yang digunakan yaitu versi HTTP/1.1



IP Address dari client yaitu 145.254.160.237

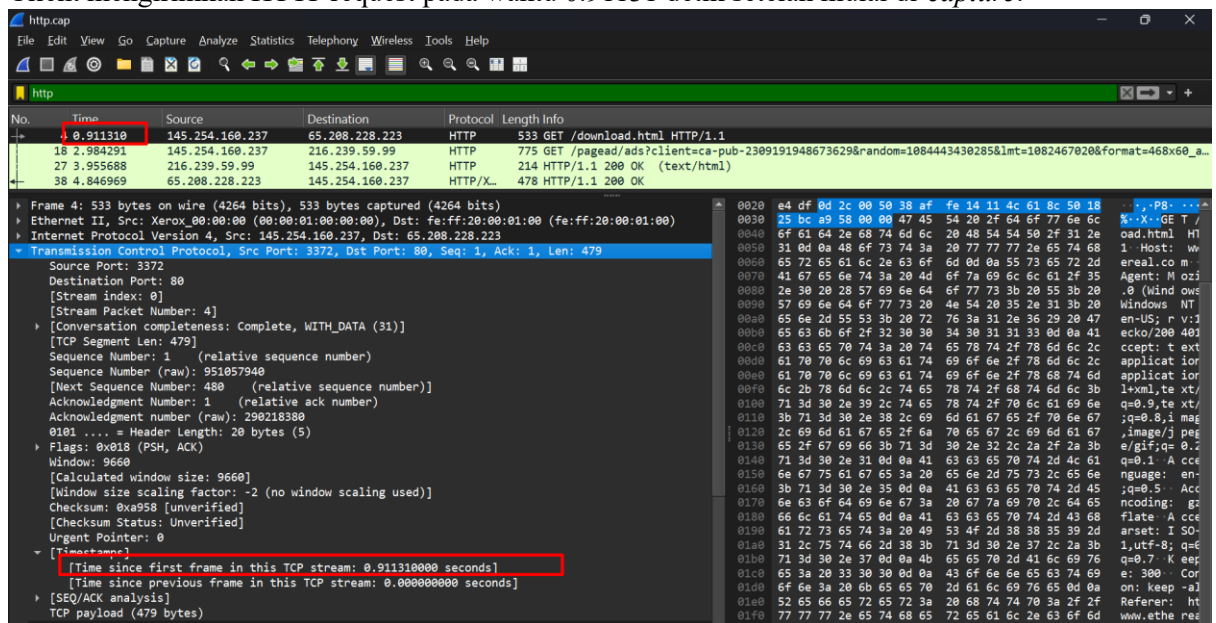


Sedangkan IP Address untuk server yaitu 65.208.228.223

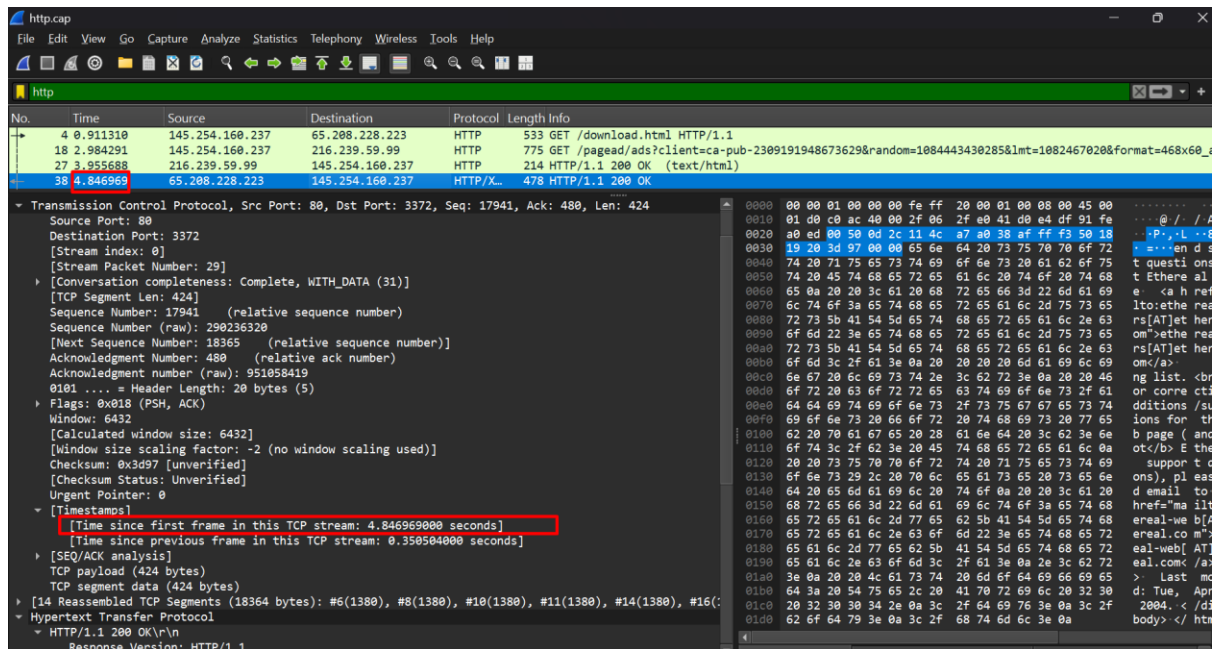


Untuk IP Address 216.239.59.99 merupakan server dari skrip iklan yang menyajikan iklan di halaman web yang dikunjungi client

Client mengirimkan HTTP request pada waktu 0.91131 detik setelah mulai di-capture.



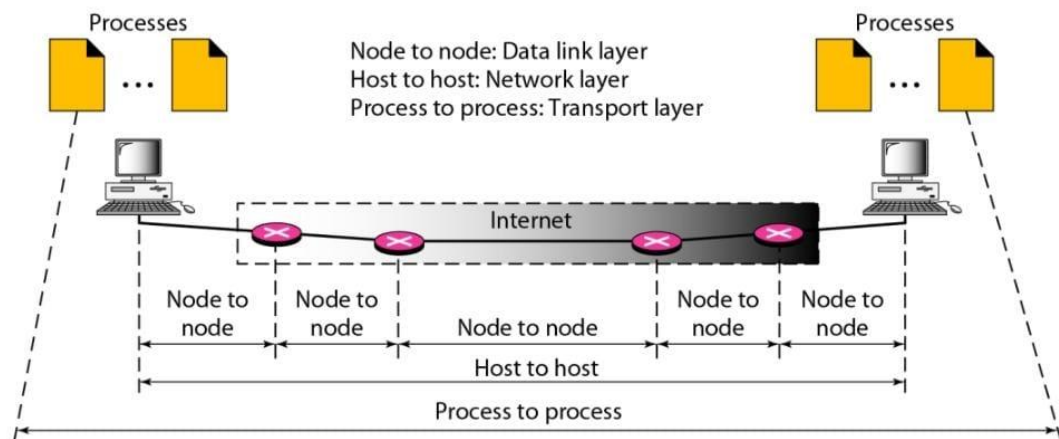
Server mengirimkan respon pada waktu 4.846969 sekon sejak wireshark mulai menangkap data



Durasi dari client meminta request sampai server memberikan OK yaitu
 $4.846969 - 0.91131 = 3.935659$ sekon

2.

Figure 23.1 *Types of data deliveries*



Ini menunjukkan bagaimana data dikirimkan.

Data link layer memiliki MAC Address agar paket bisa berpindah di dalam jaringan lokal. Jika harus keluar dari jaringan lokal, maka peran Network layer yang akan membantu paket. Di dalam network layer, ada Source IP Address dan Destination IP Address, dimana ini akan diisi alamat dari client dan server. Ketika paket keluar dari satu jaringan, MAC Address akan diganti sesuai jaringan yang ada. Ini akan terus berubah sampai paket sampai ke tujuan. Ketika sampai di tujuan, Transport layer yang akan berperan mengarahkan paket ke aplikasi yang memerlukan paket ini. Port number yang ada pada transport layer adalah bagian dari transport layer yang membantu paket harus kemana. Semisal saya membuka browser dan

mencari dua hal di 2 tab yang berbeda, maka transport layer yang memastikan bahwa paket akan terkirim ke tab yang berbeda, karena source port akan selalu berganti karena peran Dynamic Port.

3. Ada 3 tahapan utama, yaitu

1) Three-Way Handshake

Ini adalah bagaimana TCP membangun komunikasi antara client dan server. Ada 3 proses di sini, yaitu

a. SYN

Klien meminta koneksi kepada server(Synchronize)

b. SYN, ACK

Server membalas dengan respon (Synchronize-Acknowledge)

c. ACK

Klien mengembalikan ACK (Acknowledge)

Di bawah ini adalah contoh bagaimana three-way handshake berjalan

847 9.054470	192.168.1.24	18.67.175.116	TCP	66 55734 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
848 9.077317	18.67.175.116	192.168.1.24	TCP	66 443 → 55734 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM WS=512
849 9.077441	192.168.1.24	18.67.175.116	TCP	54 55734 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0

Terlihat bahwa klien dengan IP Address 192.168.1.24 meminta koneksi dengan 18.67.175.116 yang merupakan IP Address server. Server mengembalikan respon dengan SYN, ACK dan klien membalas lagi dengan ACK

2) Data Transfer

Data mulai dikirimkan dengan TCP, yang berarti data harus sampai sama persis dengan urut. Data juga tidak bisa langsung dikirim secara utuh, sehingga data akan disegmentasi dan diberi sequence number agar ketika sampai di klien, data akan diurutkan sekalipun tidak datang berurutan sehingga data akan tetap utuh.

863 9.102757	192.168.1.24	18.67.175.116	TCP	1494 55734 → 443 [ACK] Seq=3127
864 9.102757	192.168.1.24	18.67.175.116	TLSv1.3	770 Application Data
Internet Protocol Version 4, Src: 192.168.1.24, Dst: 18.67.175.116				
Transmission Control Protocol, Src Port: 55734, Dst Port: 443, Seq: 3127, Ack: 235, Len: 1440				
Source Port: 55734				
Destination Port: 443				
[Stream index: 41]				
[Stream Packet Number: 17]				
[Conversation completeness: Complete, WITH_DATA (31)]				
[TCP Segment Len: 1440]				
Sequence Number: 3127 (relative sequence number)				
Sequence Number (raw): 3503547373				
[Next Sequence Number: 4567 (relative sequence number)]				

864 9.102757	192.168.1.24	18.67.175.116	TLSv1.3	779 Application Data
865 9.102796	192.168.1.24	18.67.175.116	TCP	1494 55734 → 443 [ACK] Seq=5292
866 9.102796	192.168.1.24	18.67.175.116	TLSv1.3	1122 Application Data
867 9.102831	192.168.1.24	18.67.175.116	TCP	1494 55734 → 443 [ACK] Seq=7800
Internet Protocol Version 4, Src: 192.168.1.24, Dst: 18.67.175.116				
Transmission Control Protocol, Src Port: 55734, Dst Port: 443, Seq: 4567, Ack: 235, Len: 725				
Source Port: 55734				
Destination Port: 443				
[Stream index: 41]				
[Stream Packet Number: 18]				
[Conversation completeness: Complete, WITH_DATA (31)]				
[TCP Segment Len: 725]				
Sequence Number: 4567 (relative sequence number)				
Sequence Number (raw): 3503548813				
[Next Sequence Number: 5292 (relative sequence number)]				

865	9.102796	192.168.1.24	18.67.175.116	TCP	1494	55734 → 443 [ACK] Seq=5292
866	9.102796	192.168.1.24	18.67.175.116	TLSv1.3	1122	Application Data
867	9.102796	192.168.1.24	18.67.175.116	TCP	1494	55734 → 443 [ACK] Seq=5292
▶ Internet Protocol Version 4, Src: 192.168.1.24, Dst: 18.67.175.116						
▼ Transmission Control Protocol, Src Port: 55734, Dst Port: 443, Seq: 5292, Ack: 235, Len: 1440						
Source Port: 55734						
Destination Port: 443						
[Stream index: 41]						
[Stream Packet Number: 19]						
▶ [Conversation completeness: Complete, WITH_DATA (31)]						
[TCP Segment Len: 1440]						
Sequence Number: 5292 (relative sequence number)						
Sequence Number (raw): 3503549538						
[Next Sequence Number: 6732 (relative sequence number)]						

Seperti terlihat pada 3 gambar di atas, tiap segmen punya sequence number dan sudah punya next sequence number sehingga urutannya dapat dipastikan ketika sudah diterima

3) Connection Termination

Ini adalah cara untuk mengakhiri koneksi yang biasanya menggunakan four-way handshake. Langkah-langkahnya:

- FIN → Klien mengirim paket FIN (Finish) untuk meminta menutup koneksi.
- ACK → Server mengakui permintaan dengan ACK.
- FIN → Server mengirim FIN untuk menutup koneksi dari sisi mereka.
- ACK → Klien mengonfirmasi dengan ACK, lalu koneksi benar-benar tertutup.

Dalam penerapannya, FIN dan ACK bisa dikirim secara bersamaan atau disebut FIN + ACK. Jadi, ketika klien meminta menutup koneksi, server langsung mengakui dan meminta penutupan koneksi dari klien.

902	9.169602	18.67.175.116	192.168.1.24	TCP	54	443 → 55734 [ACK] Seq=2587 Ack=15787 Win=100864 Len=0
903	9.169602	18.67.175.116	192.168.1.24	TCP	54	443 → 55734 [ACK] Seq=2587 Ack=18318 Win=106496 Len=0
904	9.169695	192.168.1.24	52.208.252.235	TCP	54	55728 → 443 [ACK] Seq=3187 Ack=4949 Win=65280 Len=0
905	9.173715	13.216.196.87	192.168.1.24	TLSv1.2	85	[TCP Previous segment not captured], Application Data
906	9.173763	192.168.1.24	13.216.196.87	TCP	66	[TCP Dup ACK 830#1] 50037 → 443 [ACK] Seq=612 Ack=40 Win=252 Len=0 SLE=253 SRE=284
907	9.184474	98.82.157.137	192.168.1.24	SSL	1474	Continuation Data
908	9.184551	192.168.1.24	98.82.157.137	TCP	54	50053 → 443 [RST, ACK] Seq=1 Ack=1421 Win=0 Len=0
909	9.185189	18.67.175.116	192.168.1.24	TLSv1.3	400	Application Data
910	9.236826	192.168.1.24	18.67.175.116	TCP	54	55734 → 443 [ACK] Seq=18318 Ack=2933 Win=64000 Len=0
911	9.264853	192.168.1.24	3.165.190.116	TLSv1.2	89	Application Data
916	9.267435	192.168.1.24	98.82.157.137	TCP	55	[TCP Retransmission] 50120 → 443 [ACK] Seq=4294967232 Ack=1 Win=253 Len=1
930	9.494578	3.165.190.116	192.168.1.24	TCP	54	443 → 50165 [ACK] Seq=1468 Ack=518 Win=162 Len=0
931	9.799738	192.168.1.24	18.154.5.194	TCP	54	50199 → 443 [FIN, ACK] Seq=1798 Ack=5373 Win=65280 Len=0

Ini adalah contoh connection termination. Saya menutup tab dari browser secara paksa sehingga muncul permintaan (RST, ACK) dari klien. Namun, mungkin karena masih ada data yang belum terkirim, server mengirim kembali data tersebut dengan IP yang berbeda namun masih dalam satu jaringan, yaitu Amazon. Ini bisa terjadi karena server belum tahu klien sudah memutuskan koneksi. Akhirnya, klienlah yang mengirimkan IN, ACK sebagai bagian dari four-way handshake agar koneksi bisa diputuskan dengan benar. Properti ini hanya bagian dari our-way handshake yang dioptimasi untuk mempercepat proses.