Práctica Bloque III

REDES Y SISTEMAS DISTRIBUIDOS

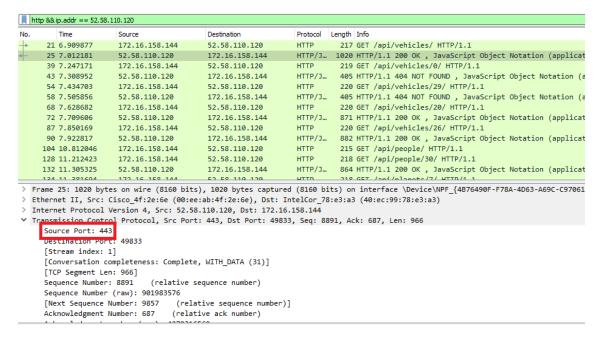
NOMBRE: Emilio Gómez Esteban

TITULACIÓN: DOBLE GRADO MATEMÁTICAS + ING INFORMÁTICA | GRUPO: D

Tarea 2: Análisis de tráfico HTTP/1.1 generado por la aplicación:

Ejercicio 1. ¿Cuál es el puerto utilizado por el servidor? ¿Es el normal de HTTP (80)? ¿Por qué?

Nº trama analizada: 25.



El puerto usado por el servidor es 443. En este caso, no nos encontramos ante el normal de HTTP, el cual es el 80, ya que estamos usando el protocolo HTTPS en lugar del habitual HTTP.

Ejercicio 2. Observe el número de conexiones realizadas. ¿Cuántas hace? ¿Usa una conexión permanente (en la misma conexión hace varias peticiones) o no permanente (solo realiza una por conexión)? En caso de ser permanente, ¿qué cabecera de la petición indica que queremos que sea permanente?

Nº tramas analizadas: 21, 25, 436.

	http && ip.addr == 52.58.110.120							
No.	Time	Source	Destination	Protocol	Length Info			
	21 6.909877	172.16.158.144	52.58.110.120	HTTP	217 GET /api/vehicles/ HTTP/1.1			
4	25 7.012181	52.58.110.120	172.16.158.144	HTTP/J	. 1020 HTTP/1.1 200 OK , JavaScript Object Notation (appli			
	39 7.247171	172.16.158.144	52.58.110.120	HTTP	219 GET /api/vehicles/0/ HTTP/1.1			
	43 7.308952	52.58.110.120	172.16.158.144	HTTP/J	. 405 HTTP/1.1 404 NOT FOUND , JavaScript Object Notation			
	54 7.434703	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/29/ HTTP/1.1			
	58 7.505856	52.58.110.120	172.16.158.144	HTTP/J	. 405 HTTP/1.1 404 NOT FOUND , JavaScript Object Notation			
	68 7.628682	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/20/ HTTP/1.1			
	72 7.709606	52.58.110.120	172.16.158.144	HTTP/J	. 871 HTTP/1.1 200 OK , JavaScript Object Notation (appli			
	87 7.850169	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/26/ HTTP/1.1			
	90 7.922817	52.58.110.120	172.16.158.144	HTTP/J	. 882 HTTP/1.1 200 OK , JavaScript Object Notation (appli			
	104 10.812046	172.16.158.144	52.58.110.120	HTTP	215 GET /api/people/ HTTP/1.1			
	128 11.212423	172.16.158.144	52.58.110.120	HTTP	218 GET /api/people/30/ HTTP/1.1			
	132 11.305325	52.58.110.120	172.16.158.144	HTTP/J	. 864 HTTP/1.1 200 OK , JavaScript Object Notation (appli			
	124 11 201604	170 16 150 144	E2 E0 110 120	HTTD	210 GET /ani/planate/7/ HTTD/1 1			
>	> Frame 21: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{4876490F-F78A-4D63-A69C-C97061							
>	> Ethernet II, Src: IntelCor_78:e3:a3 (40:ec:99:78:e3:a3), Dst: Cisco_4f:2e:6e (00:ee:ab:4f:2e:6e)							
>	> Internet Protocol Version 4, Src: 172.16.158.144, Dst: 52.58.110.120							
~	▼ Tr <u>ansmission Control P</u> rotocol, Src Port: 49833, Dst Port: 443, Seq: 524, Ack: 4511, Len: 163							
	Source Port: 49833							
	Destination Port	: 443						
	[Stream index: 1]							
	[Conversation completeness: Complete, WITH_DATA (31)]							
	[TCP Segment Len: 163]							
	Sequence Number: 524 (relative sequence number)							
	Sequence Number (raw): 4270316406							
	[Next Sequence N	Number: 687 (relat:	ive sequence number)]					
	Acknowledgment Number: 4511 (relative ack number)							
	4 1 1 1 1 / \ 000070406							

lo.	Time	Source	Destination	Protocol	Length Info				
٠.	351 21.294542	172.16.158.144	52.58.110.120	HTTP	219 GET /api/planets/55/ HTTP/1.1				
	353 21.371638	52.58.110.120	172.16.158.144	HTTP/J					
	367 21.493397	172.16.158.144	52.58.110.120	HTTP	218 GET /api/people/19/ HTTP/1.1				
	371 21.560938	52.58.110.120	172.16.158.144	HTTP/J	866 HTTP/1.1 200 OK , JavaScript Object Notation				
	372 21.576245	172.16.158.144	52.58.110.120	HTTP	219 GET /api/planets/26/ HTTP/1.1				
	374 21.649223	52.58.110.120	172.16.158.144	HTTP/J	809 HTTP/1.1 200 OK , JavaScript Object Notation				
	388 21.845387	172.16.158.144	52.58.110.120	HTTP	217 GET /api/people/0/ HTTP/1.1				
	392 21.908399	52.58.110.120	172.16.158.144	HTTP/J	405 HTTP/1.1 404 NOT FOUND , JavaScript Object				
	402 22.031334	172.16.158.144	52.58.110.120	HTTP	218 GET /api/people/26/ HTTP/1.1				
	406 22.124434	52.58.110.120	172.16.158.144	HTTP/J	the state of the s				
	407 22.129448	172.16.158.144	52.58.110.120	HTTP	218 GET /api/planets/6/ HTTP/1.1				
	409 22.226880	52.58.110.120	172.16.158.144	HTTP/J	the state of the s				
	425 22.349242	172.16.158.144	52.58.110.120	HTTP	218 GET /api/people/60/ HTTP/1.1				
	433 22.416539	52.58.110.120	172.16.158.144	НТТР/Ј	to the territory of the				
	434 22.420998	172.16.158.144	52.58.110.120	HTTP	218 GET /api/planets/8/ HTTP/1.1				
-	436 22.517989	52.58.110.120	172.16.158.144	НТТР/Ј	1308 HTTP/1.1 200 OK , JavaScript Object Notatio				
	\								
> Frame 436: 1308 bytes on wire (10464 bits), 1308 bytes captured (10464 bits) on interface \Device\NPF_{4876490F-F78A-4} > Ethernet II, Src: Cisco_4f:2e:6e (00:ee:ab:4f:2e:6e), Dst: IntelCor_78:e3:a3 (40:ec:99:78:e3:a3)									
> Internet Protocol Version 4, Src: 52.58.110.120, Dst: 172.16.158.144									
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 49853, Seq: 945, Ack: 773, Len: 1254 Source Port: 443									
Destination Port: 49853									
Stream index: 21									
[Conversation completeness: Complete, WITH DATA (31)]									
[TCP Segment Len: 1254]									

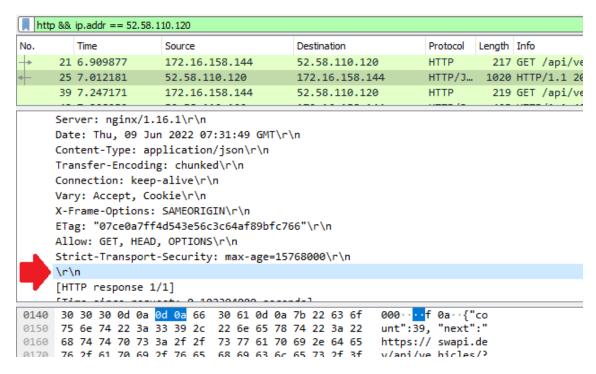
Para calcular el número de conexiones usadas, veremos el número de puertos que usan HTTP para conversar con el servidor. Restando el último y el primero usado, obtenemos que se han usado 20 conexiones.

ı	http && ip.addr == 52.58.110.120							
No.	Time	Source	Destination	Protocol	Length Info			
-	21 6.909877	172.16.158.144	52.58.110.120	HTTP	217 GET /api/vehicles/ HTTP/1.1			
4	25 7.012181	52.58.110.120	172.16.158.144	НТТР/Ј	1020 HTTP/1.1 200 OK , JavaScript Object Notation (application)			
	39 7.247171	172.16.158.144	52.58.110.120	HTTP	219 GET /api/vehicles/0/ HTTP/1.1			
	43 7.308952	52.58.110.120	172.16.158.144	HTTP/J	405 HTTP/1.1 404 NOT FOUND , JavaScript Object Notation (ap			
	54 7.434703	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/29/ HTTP/1.1			
	58 7.505856	52.58.110.120	172.16.158.144	HTTP/J	405 HTTP/1.1 404 NOT FOUND , JavaScript Object Notation (ap			
	68 7.628682	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/20/ HTTP/1.1			
	72 7.709606	52.58.110.120	172.16.158.144	HTTP/J	871 HTTP/1.1 200 OK , JavaScript Object Notation (applicati			
	87 7.850169	172.16.158.144	52.58.110.120	HTTP	220 GET /api/vehicles/26/ HTTP/1.1			
	90 7.922817	52.58.110.120	172.16.158.144	HTTP/J	882 HTTP/1.1 200 OK , JavaScript Object Notation (applicati			
	104 10.812046	172.16.158.144	52.58.110.120	HTTP	215 GET /api/people/ HTTP/1.1			
	128 11.212423	172.16.158.144	52.58.110.120	HTTP	218 GET /api/people/30/ HTTP/1.1			
	132 11.305325	52.58.110.120	172.16.158.144	HTTP/J	(-FF			
		170 16 150 144	E2 E0 110 120	UTTD	219 GET /ppi/planote/7/ HTTD/1 1			
	Frame 25: 1020 bytes on wire (8160 bits), 1020 bytes captured (8160 bits) on interface \Device\NPF_{4B76490F-F78A-4D63-A69C-C97061							
	•	_ ,	, , ,	_	:e3:a3 (40:ec:99:78:e3:a3)			
	Internet Protocol Version 4, Src: 52.58.110.120, Dst: 172.16.158.144							
>	Transmission Control Protocol, Src Port: 443, Dst Port: 49833, Seq: 8891, Ack: 687, Len: 966							
>	[3 Reassembled TCP Segments (5346 bytes): #23(1460), #24(2920), #25(966)]							
	Transport Layer Sec							
~	Hypertext Transfer							
	> HTTP/1.1 200 OK\r\n							
	Server: nginx/1.16.1\r\n							
	Date: Thu, 09 Jun 2022 07:31:49 GMT\r\n							
	Content-Type: application/json\r\n							
	Transfer-Encoding: chunked\r\n							
	Connection: keep-alive\r\n							

Usa una conexión permanente, porque indica el nombre de la conexión como vemos en la imagen anterior: "keep-alive".

Ejercicio 3. Observe una respuesta, ¿cómo se identifica dónde acaban las cabeceras HTTP y empieza el recurso?

Nº trama analizada: 25.



Se identifica con una línea que contiene la cadena "\r\n", la cual no tiene más información y separa el recurso de las cabeceras. Justo después de esta línea comienza el recurso.

Ejercicio 4. Describa el significado de las cabeceras de una petición y una respuesta (sin incluir las que empiecen por X-).

PETICIÓN:

Hypertext Transfer Protocol > GET /api/vehicles/ HTTP/1.1\r\n User-Agent: My SW Client v0.1-2022\r\n Accept: application/json\r\n Host: swapi.dev\r\n Connection: keep-alive\r\n \r\n

- 1.GET /api/vehicles/HTTP/1.1: tipo de petición, dirección y versión de HTTP.
- 2.User-Agent: My SW Client v0.1-2022: nombre de la aplicación.
- 3. Accept: application/json: formato de la información.
- 4. Host: swapi.dev: nombre de host
- 5. Connection: keep-alive: tipo de conexión.

RESPUESTA:

```
Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Server: nginx/1.16.1\r\n

Date: Thu, 09 Jun 2022 07:31:49 GMT\r\n

Content-Type: application/json\r\n

Transfer-Encoding: chunked\r\n

Connection: keep-alive\r\n

Vary: Accept, Cookie\r\n

X-Frame-Options: SAMEORIGIN\r\n

ETag: "07ce0a7ff4d543e56c3c64af89bfc766"\r\n

Allow: GET, HEAD, OPTIONS\r\n

Strict-Transport-Security: max-age=15768000\r\n
\r\n
```

- 1.HTTP/1.1 200 OK: versión de HTTP y estado conexión.
- 2. Server: nginx/1.16.1: servidor.
- 3.Date: Thu, 09 Jun 2022 07:31:49 GMT: fecha y hora de petición.
- 4. Content-Type: application/json: formato contenido.
- 5. Transfer-Encoding: chunked: envío realizador mediante chunks separados.

6.Connection: keep-alive: tipo de conexión.

7. Vary: Accept, Cookie: parámetros de la respuesta.

8.ETag: "07ce0a7ff4d543e56c3c64af89bfc766: identificador.

9.Allow: GET, HEAD, OPTIONS: métodos.

10.Strict-Transport-Security: max-age=15768000: tiempo de vida.