Práctica Bloque III

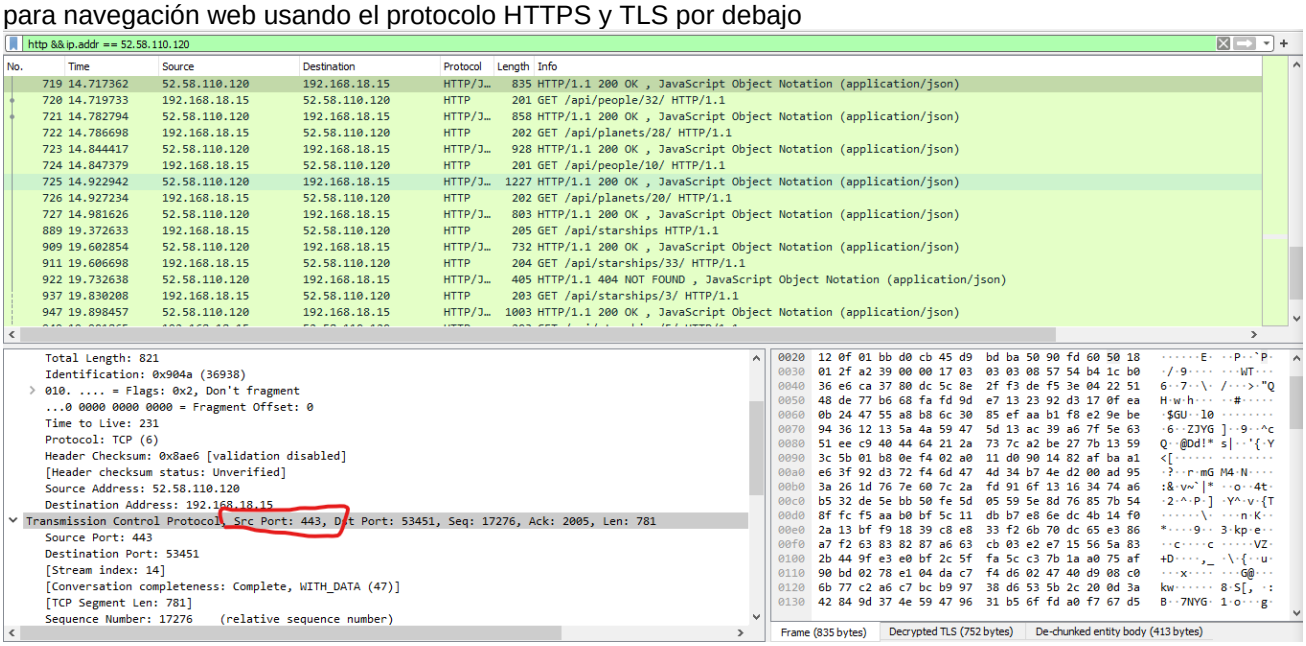
Alumno: Navarro Jimena, David

Titulación: Grado de Ingeniería Informática

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

- Tramas analizadas: 719

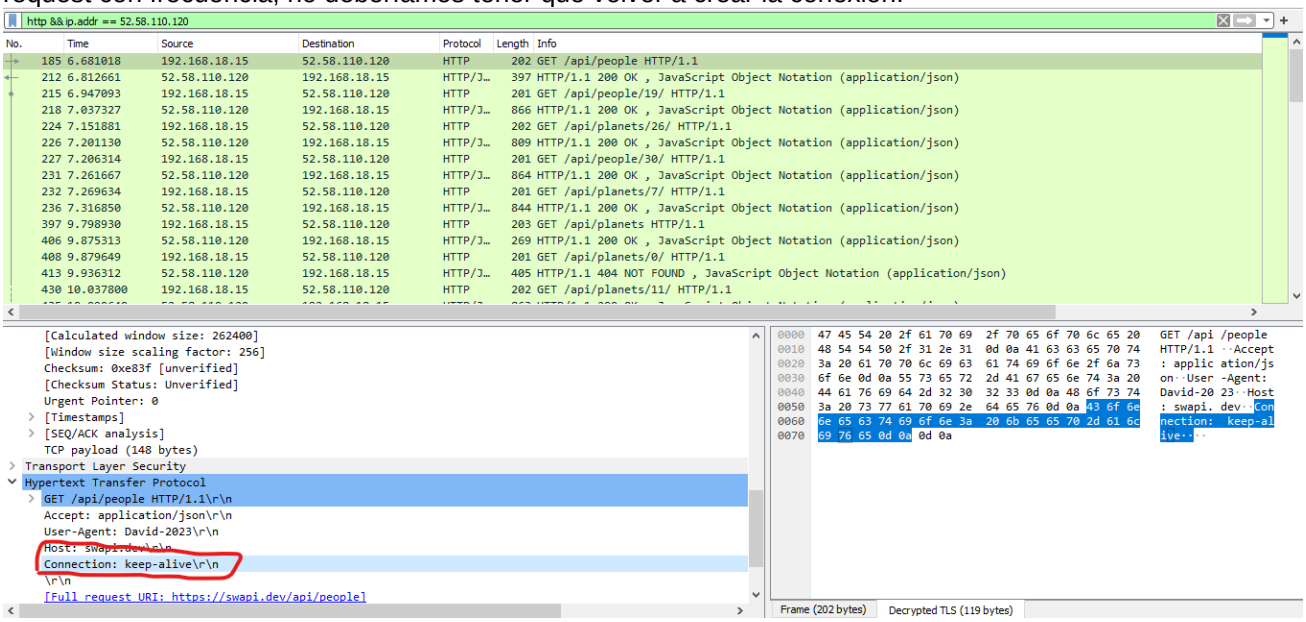
Utiliza el puerto 443, uno distinto al normal de HTTP porque usa HTTPS. El puerto 443 es un puerto también para navegación web usando el protocolo HTTPS y TLS por debajo



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?

- Tramas analizadas:719

Realiza una sola conexión. Usa una conexión permanente al hacer varias request por conexión. Para que sea permanente tenemos que añadir en la cabecera el argumento keep-alive. Así que mientras hagamos request con frecuencia, no deberíamos tener que volver a crear la conexión.



Ejercicio 3. Describa el significado de las cabeceras de una petición y una respuesta (sin incluir las `X-*`).

- Tramas analizadas:719

```
GET /api/people HTTP/1.1
Accept: application/json
User-Agent: David-2023
Host: swapi.dev
Connection: keep-alive

HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Tue, 30 May 2023 14:44:20 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
ETag: "b493126da505af6fec015ec116fec193"
Allow: GET, HEAD, OPTIONS
Strict-Transport-Security: max-age=15768000
```

En rojo tenemos la cabecera de la petición donde vemos el método HTTP, en este GET, y a que llamamos. La versión de HTTP, que tipo de datos aceptamos con Accept, en este caso json, el usuario que hace la llamada, el host y por último, que queremos mantener la conexión viva.

En azul tenemos la respuesta. Donde podemos ver primero la versión de HTTP y estado de la request. Es este 200 así que todo bien. Después vemos el servidor, la fecha, el tipo de contenido que tiene que ser igual al pedido por la request. El tipo de codificación para la transmisión y el argumento de la conexión. Vary sirve para hacer coincidir los encabezados de solicitudes futuras para ver si se puede usar la cache. Tras esto tenemos X-Frame-Options es una opción para saber si el navegador debería renderizar esos datos, en este caso solo se mostraría si todos los frames tienen el mismo origen que la página. Ahora tenemos ETag que es un identificador para una versión del recurso y que la cache sea más eficiente. Por último tenemos Allow que nos dice que métodos HTTP están permitidos y Strict-Transport-Security donde podemos poner opciones para el transporte

Ejercicio 4. Filtre por el protocolo `rtsp` y use la opción **Follow TCP Stream** de Wireshark para observar el diálogo completo que han mantenido el cliente de correo y el servidor. Explique brevemente (una línea) el significado de cada comando enviado por el cliente (si algún comando se repite solo debe explicarlo una vez).

- Tramas analizadas:8,11,23,37,1617,6727

```
OPTIONS rtsp://rtsp.stream:554/movie RTSP/1.0
CSeq: 2
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
```

Con options podemos indicar la URI del archivo multimedia que queremos

```
DESCRIBE rtsp://rtsp.stream:8554/bunny.mkv RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Accept: application/sdp
```

Con este método podemos conseguir una descripción del archivo

```
SETUP rtsp://23.88.67.97:8554/bunny.mkv/track1 RTSP/1.0
CSeq: 5
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=51500-51501
```

Con este comando podemos indicar el tipo de mecanismo de transporte para el archivo indicado

```
PLAY rtsp://23.88.67.97:8554/bunny.mkv/ RTSP/1.0
CSeq: 7
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1352FCCC
Range: npt=0.000-
```

Con este indicamos al servidor que comience a enviar los datos

```
PAUSE rtsp://23.88.67.97:8554/bunny.mkv/ RTSP/1.0
CSeq: 8
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1352FCCC
```

Con este pausamos la transmisión

```
TEARDOWN rtsp://23.88.67.97:8554/bunny.mkv/ RTSP/1.0
CSeq: 12
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1352FCCC
```

Y por último, con teardown hacemos que pare la transmisión

Ejercicio 5. ¿Por qué se hacen dos comandos `SETUP`? ¿Cómo sabía que debía hacer dos comandos de ese estilo?

- Tramas analizadas:23,26,20

Realiza dos comandos SETUP para indicar los parámetros de transporte de cada una de las tracks del video, además de indicar en el segundo la sesión que está usando el cliente. Sabe que tiene que hacer eso ya que en DESCRIBE se muestra que hay 2 tracks.

```
v=0
o=- 1654711921171645 1 IN IP4 23.88.67.97
s=Matroska video+audio+(optional)subtitles, streamed by RTSP.Stream
i=bunny.mkv
t=0 0
a=tool:LIVE555 Streaming Media v2021.11.01
a=type:broadcast
a=control:*
a=range:npt=0-634.642
a=x-qt-text-nam:Matroska video+audio+(optional)subtitles, streamed by RTSP.Stream
a=x-qt-text-inf:bunny.mkv
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
b=AS:500
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=64001F;sprop-parameter-sets=Z2QAH6zZgUH7DmoCAgKAAAADAIABX5AHjBjN,aOl5yyLA
a=control:track1
m=audio 0 RTP/AVP 97
c=IN IP4 0.0.0.0
b=AS:96
a=rtpmap:97 MPEG4-GENERIC/48000
a=fmtp:97 streamtype=5;profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalength=3;config=118856E500
a=control:track2
```

Ejercicio 6. ¿Qué comandos ha provocado adelantar la reproducción del vídeo? ¿Cómo indica por donde debe seguir la reproducción tras el cambio?

- Tramas analizadas: 2945, 2927

Tiene primero que pausar el video y después usar PLAY indicando en el argumento Range el momento del video que queremos.

```
PAUSE rtsp://23.88.67.97:8554/bunny.mkv/ RTSP/1.0
CSeq: 10
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1352FCCC

RTSP/1.0 200 OK
CSeq: 10
Date: Wed, Jun 08 2022 18:12:32 GMT
Session: 1352FCCC

PLAY rtsp://23.88.67.97:8554/bunny.mkv/ RTSP/1.0
CSeq: 11
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1352FCCC
Range: npt=342.199-
```

Ejercicio 7. Si observa los comandos y las respuestas son muy similares a las que usa HTTP. Indique dos cabeceras que use RTSP que también se usen en HTTP e indique (y explique) dos cabeceras de RTSP que no se usen en HTTP.

- Tramas analizadas:

En ambos protocolos se usan User-Agent y Accept. Pero en RTSP tenemos Transport que tenemos que usar para indicar el protocolo y el puerto, cosa que no necesitamos en HTTP. Y también tenemos CSeq para indicar el par de respuesta y request para poder saber cuáles van juntas, así aunque lleguen desordenadas se puede saber que respuestas pertenecen a que request.

Ejercicio 8. Ahora filtre por el protocolo `rtp` que se utiliza para transmitir el recurso multimedia tal cual. ¿Cómo se decidieron los puertos a utilizar en estas comunicaciones RTP? ¿Se confirman de alguna forma cada uno de los envíos RTP?

- Tramas analizadas:43

Se deciden al usar el comando SETUP. No, no se confirman, funciona como una conexión UDP, sería muy lento si se tuvieran que confirmar.