

PRÁCTICA 04

Protocolo HTTP

Alexis Coves
Berna DAW
2ºW Grupo 2

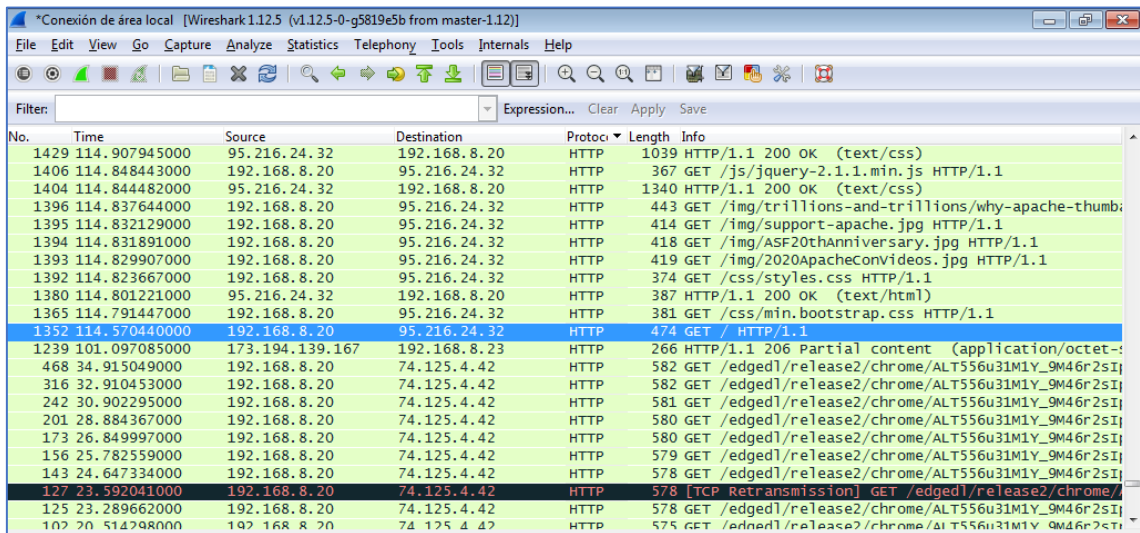
Índice:

1. Protocolo HTTP	2
2. Análisis	3
3. Herramientas para desarrolladores de Google Chrome	4
4. Cookies	5

1. Protocolo HTTP:

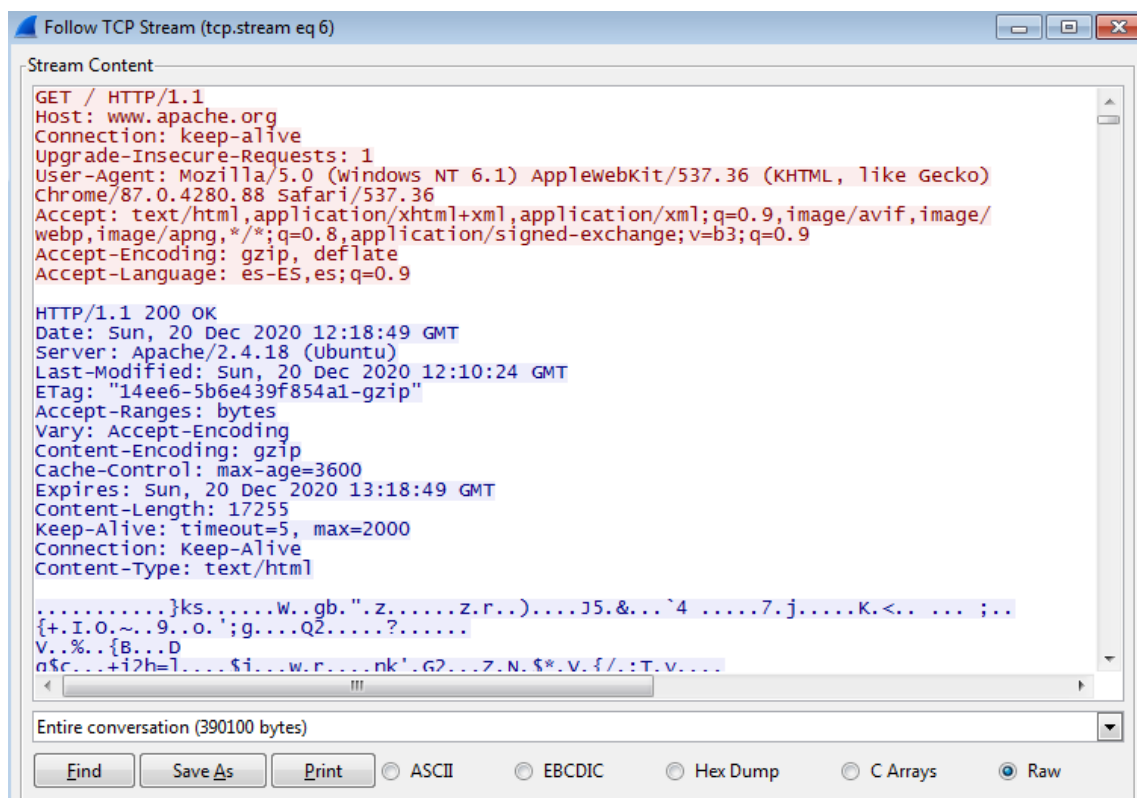
En esta práctica vamos a analizar la información de los mensajes de petición y respuesta del protocolo HTTP.

Para ello, abriremos el navegador Google Chrome y estableceremos una conexión con "http://www.apache.org". A continuación, iniciaremos una captura con el software de análisis de protocolos Wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
1429	114.907945000	95.216.24.32	192.168.8.20	HTTP	1039	HTTP/1.1 200 OK (text/css)
1406	114.848443000	192.168.8.20	95.216.24.32	HTTP	367	GET /js/jquery-2.1.1.min.js HTTP/1.1
1404	114.844482000	95.216.24.32	192.168.8.20	HTTP	1340	HTTP/1.1 200 OK (text/css)
1396	114.837644000	192.168.8.20	95.216.24.32	HTTP	443	GET /img/trillions-and-trillions/why-apache-thumb...
1395	114.832129000	192.168.8.20	95.216.24.32	HTTP	414	GET /img/support-apache.jpg HTTP/1.1
1394	114.831891000	192.168.8.20	95.216.24.32	HTTP	418	GET /img/ASF20thAnniversary.jpg HTTP/1.1
1393	114.829907000	192.168.8.20	95.216.24.32	HTTP	419	GET /img/2020ApacheConVideos.jpg HTTP/1.1
1392	114.823667000	192.168.8.20	95.216.24.32	HTTP	374	GET /css/styles.css HTTP/1.1
1380	114.801221000	95.216.24.32	192.168.8.20	HTTP	387	HTTP/1.1 200 OK (text/html)
1365	114.791447000	192.168.8.20	95.216.24.32	HTTP	381	GET /css/min.bootstrap.css HTTP/1.1
1352	114.570440000	192.168.8.20	95.216.24.32	HTTP	474	GET / HTTP/1.1
1239	101.097085000	173.194.139.167	192.168.8.23	HTTP	266	HTTP/1.1 206 Partial content (application/octet-...
468	34.915049000	192.168.8.20	74.125.4.42	HTTP	582	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
316	32.910453000	192.168.8.20	74.125.4.42	HTTP	582	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
242	30.902295000	192.168.8.20	74.125.4.42	HTTP	581	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
201	28.884367000	192.168.8.20	74.125.4.42	HTTP	580	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
173	26.849997000	192.168.8.20	74.125.4.42	HTTP	580	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
156	25.782559000	192.168.8.20	74.125.4.42	HTTP	579	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
143	24.647334000	192.168.8.20	74.125.4.42	HTTP	578	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
127	23.592041000	192.168.8.20	74.125.4.42	HTTP	578	[TCP Retransmission] GET /edgedl/release2/chrome/...
125	23.289662000	192.168.8.20	74.125.4.42	HTTP	578	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...
102	20.512248000	192.168.8.20	74.125.4.42	HTTP	575	GET /edgedl/release2/chrome/ALT556u31M1Y_9M46r2sI...

Localizamos la trama en la que la petición sea "GET / HTTP/1.1"



Follow TCP Stream (tcp.stream eq 6)

Stream Content

```

GET / HTTP/1.1
Host: www.apache.org
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9

HTTP/1.1 200 OK
Date: Sun, 20 Dec 2020 12:18:49 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 20 Dec 2020 12:10:24 GMT
ETag: "14ee6-5b6e439f854a1-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Cache-Control: max-age=3600
Expires: Sun, 20 Dec 2020 13:18:49 GMT
Content-Length: 17255
Keep-Alive: timeout=5, max=2000
Connection: Keep-Alive
Content-Type: text/html
  
```

Entire conversation (390100 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Una vez localizada haremos click derecho y seleccionamos la opción "Follow TCP Stream".

2. Análisis:

El *stream* de la petición nos muestra un resumen con una serie de preguntas y respuestas realizadas en la conexión además de sus metadatos. Podemos distinguir las preguntas representadas con un color rojo y las respuestas con un color azul.

Como podemos observar, la dirección IP en la cual se ejecuta el servidor web es "95.216.24.32". Esta dirección podría no ser siempre la misma ya que puede haber distintos servidores web.

Comprobamos que utiliza la versión 1.1 de HTTP, es decir la versión estándar más utilizada actualmente.

Observamos que el método de petición es de tipo *GET*, es decir que los datos son visibles en la URL.

El recurso que solicita la petición es el directorio raíz del host "www.apache.org".

No encontramos cookies en la petición utilizando la herramienta *find* de wireshark en el *stream* de la petición.

El navegador utiliza el lenguaje es-ES.

Observamos que el código de respuesta es el 200, lo cual indica que la respuesta ha sido correcta.

Observamos que el servidor web es apache en su versión 2.4.29.

Observamos que el tipo de MIME (*Multipurpose Internet Maill Extensions*) es *text/html*, esto quiere decir que representa un texto legible por humanos y además es un documento HTML.

Comprobamos que se han realizado múltiples preguntas y respuestas por lo que se han utilizado conexiones persistentes, éstas permanecen abiertas hasta que el cliente o el servidor deciden cerrarlas, lo que permite que puedan realizarse diversas preguntas y respuestas sin la necesidad de abrir y cerrar conexiones continuamente, esto mejora considerablemente el rendimiento del protocolo HTTP.

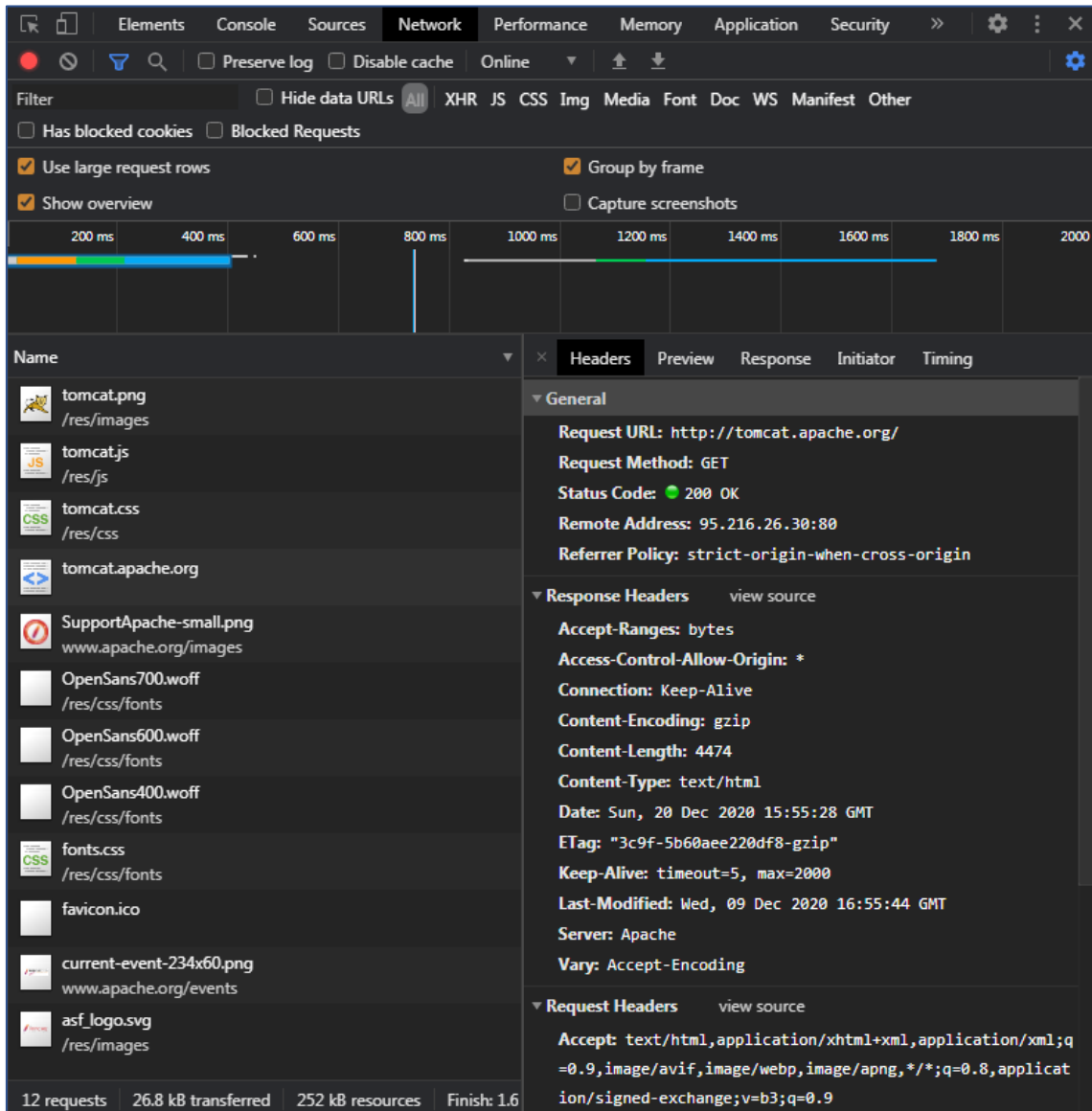
Comprobamos que hay peticiones y respuestas de imágenes siguiendo el estándar MIME. Encontramos imágenes del tipo *img/png* y *img/vnd.microsoft.icon*.

```
Last-Modified: Mon, 12 Aug 2019 09
ETag: "4f84-58fe7da7b8f8a"
Accept-Ranges: bytes
Content-Length: 20356
Content-Type: image/png
```

```
Cache-Control: max-age=3600
Expires: Sun, 20 Dec 2020 14:37:00 GMT
Keep-Alive: timeout=5, max=1995
Connection: Keep-Alive
Content-Type: image/vnd.microsoft.icon
```

3. Herramientas para desarrolladores de Google Chrome:

Analizamos las peticiones de la conexión con `http://tomcat.apache.org/` mediante la herramienta para desarrolladores de Google Chrome.

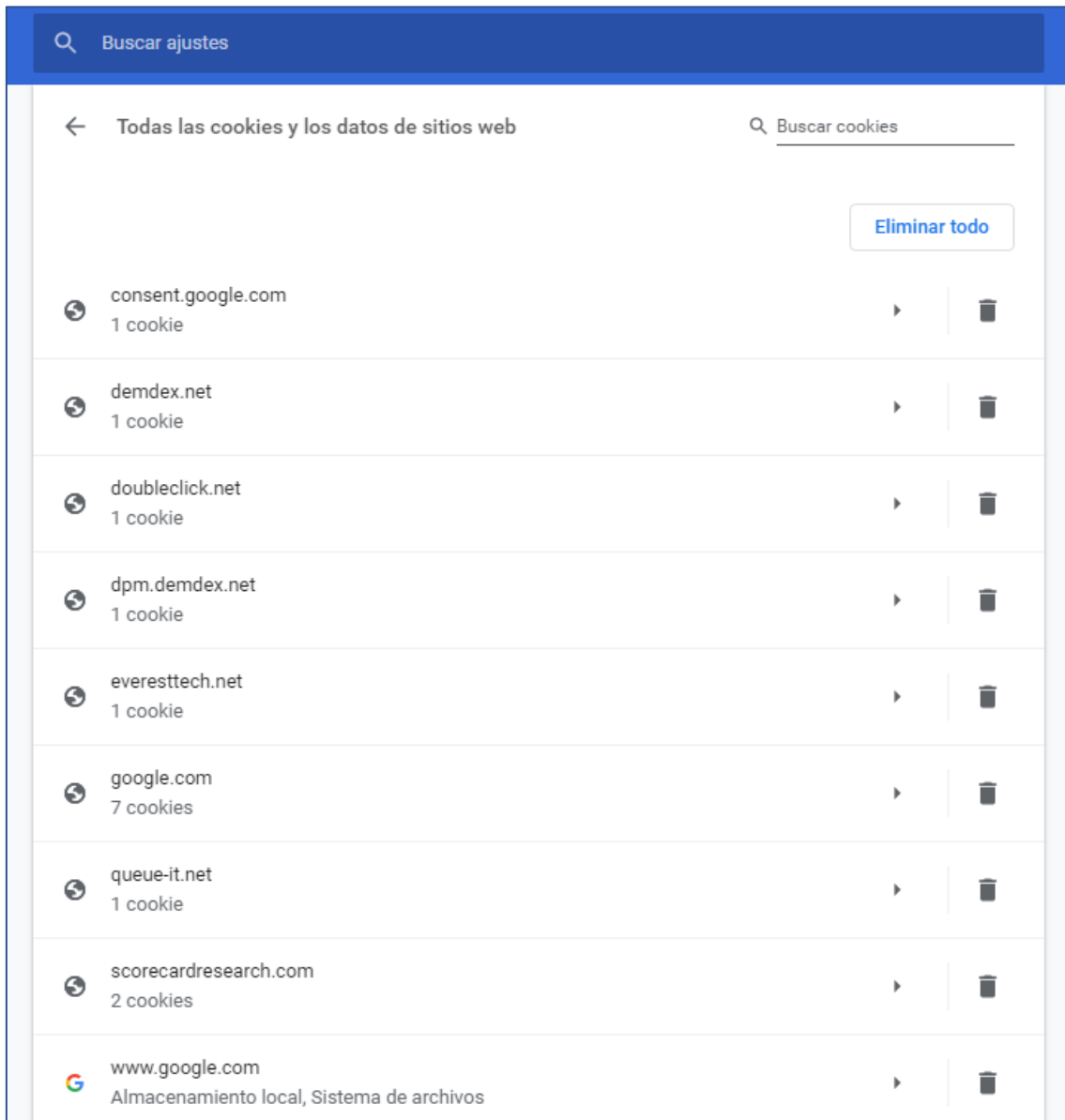


Abrimos las herramientas para desarrolladores de Google Chrome.

Comprobamos que todas las peticiones usan el método `GET` y los códigos de respuesta son todos 200, por lo que todas se han ejecutado correctamente. Observamos que los recursos que envía el servidor son elementos que forman la estructura del DOM. Encontramos peticiones para el HTML, CSS, JavaScript, imágenes, fuentes etc.

4. Cookies

Desde el navegador podemos observar las cookies que tenemos almacenadas, estas cookies a menudo guardan información relevante para el servidor, como por ejemplo detectar si es un usuario nuevo, o si ya está registrado, recordar sus datos de inicio de sesión, aunque también pueden usarse por agencias de publicidad para conocer los hábitos de navegación de los usuarios pudiendo causar problemas de privacidad.



Comprobamos las cookies almacenadas en el navegador.