Práctica 3: HTTP

1. Comunicación cliente-servidor HTTP

1.1 La **dirección IP** de la máquina cliente HTTP es **13.0.0.13** y la del servidor es **21.0.0.21**.

1.2 La **versión HTTP** que se utiliza en esta comunicación es la **1.1**.

1.3 Solo se ve una única conexión, ya que no en ningún momento el servidor envía un mensaje TCP para indicar el cierre de la conexión. Los recursos del mismo servidor se transfieren todos por la misma conexión TCP al haber una sola conexión en la captura.

1.4 Hay 2 peticiones **GET** desde el cliente.

1.5 El usuario ha escrito una **URL** en el navegador para obtener dicha captura, ya que la segunda petición **GET** se trata de una imagen proveniente de la misma **URL**, **http://www1/foto1.jpg**.
La URL escrita es **http://www1/index.html.**

1.6 Le enviará un GET posteriormente para solicitar la imagen que se encuentra en ese dominio.

1.7 La **versión HTTP** que se utiliza en esta comunicación es la **1.0**.

1.8 Hay una única conexión en el fichero de captura, siendo la utilizada para la transferencia de recursos del mismo servidor. La diferencia con esta captura y en la anterior, es que en la anterior captura la conexión continuaba abierta y en esta se cierra.

2. Diferentes tipos de respuestas de un servidor

2.1 El contenido del campo "**Estado**" es ***404 Not Found***, tratándose de un código de estado que nos indica que ha habido un error en el lado del cliente.

Los otros **GET** realizados se tratan de peticiones de imágenes que se han procesado correctamente empleados para mostrar el mensaje de error ***404 Not Found*** gráficamente.

2.2 En el primer **GET** se indica que el recuro solicitado se ha movido definitivamente a la URL proporcionada por los encabezados de ubicación.
En el segundo **GET** la petición se ha procesado correctamente, y podemos observar que el dominio de la URL solicitada cambió de **http://www.wikipedia.com** a **http://www.wikipedia.org**.

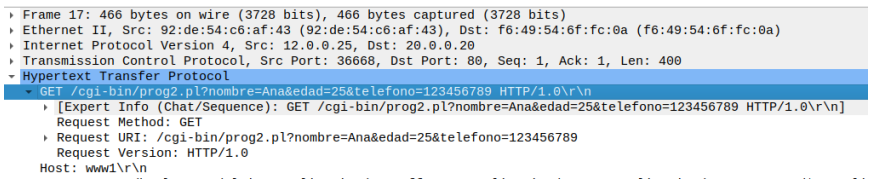
3. Formularios en HTTP

3.1 El número de conexiones en esta comunicación es 2.

3.2 Los nombres de los campos del formulario que rellenará el usuario son "*Nombre*", "*Edad*" y "*Usuario*".

3.3 Por código HTML, el método se elige rellenando el campo "***method***".
En este caso el método escogido es **GET**, dado que así aparece en la cabecera del mensaje de solicitud**.**

3.4



Mediante el campo “***Request Method***” comprueba que se está usando el método indicado en el apartado anterior.

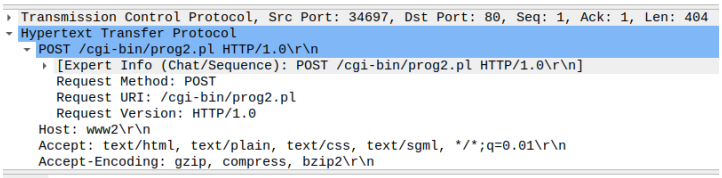
3.5 El programa del servidor que va a recibir esos datos es **Apache/1.3.33 (Debian GNU/Linux).**

3.6 Los datos que el cliente le envía al servidor viajan en el **path de la linea inicial (URL)**, dado que no hay cuerpo. Esos datos son el contenido que el cliente haya rellenado en los campos “*Nombre*”, “*Edad*” y “*Usuario*”.

3.9 Los nombres de los campos del formulario que rellenará el usuario son “*Nombre*”, “*NIF*” y “*Edad*”.

3.10 Por código HTML, el método se elige rellenando el campo “***method*”.**
En este caso el método escogido es **POST**, dado que así aparece en la cabecera del mensaje de solicitud**.**

3.11



Mediante el campo “***Request Method***” comprueba que se está usando el método indicado en el apartado anterior.

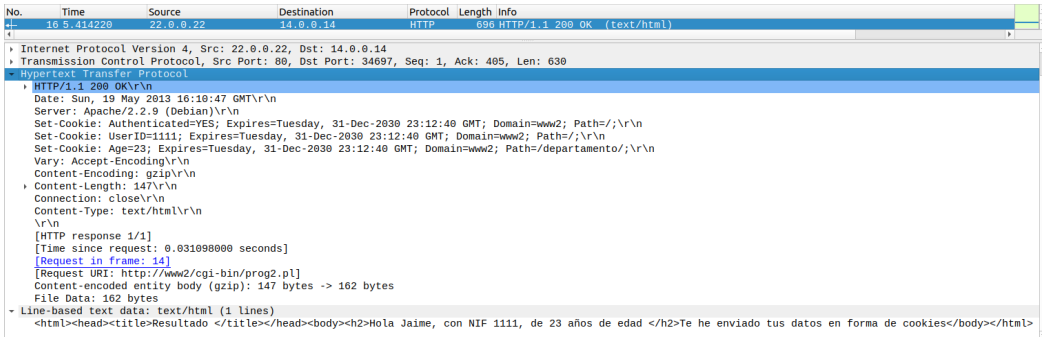
3.12 El programa del servidor que va a recibir esos datos es **Apache/2.2.9 (Debian)**.

3.13 Los datos que el cliente le envía al servidor viajan en el **cuerpo**. Esos datos son el contenido que el cliente haya rellenado en los campos “*Nombre*”, “*NIF*” y “*Edad*”.

3.14 La cabecera que representa el tipo de contenido que el cliente le envía al servidor es “***Content-Type***” siendo su valor ***application/x-www-form-urlencoded.***

3.15 Es necesario ya que el paquete contiene un cuerpo, por lo que ha de ser especificado su número de bytes mediante el campo “***Content-Length***”.

3.16



4. Cookies

4.2 Envío de cookies en mensajes HTTP

4.2.1 Las cookies que envía el servidor al cliente son las siguientes:
Set-Cookie: *Nombre=Pepe Lozano; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=elcortebritanico.com; Path=/facturas/;*

Set-Cookie: *Nif=22034J; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=elcortebritanico.com; Path=/facturas;*
Set-Cookie: *Edad=22; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=elcortebritanico.com; Path=/facturas;*
Set-Cookie: *Carrito=obsequio-bienvenida; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=elcortebritanico.com; Path=/tienda;*
Set-Cookie: *Sesion=1003; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=elcortebritanico.com; Path=/;*

4.2.2 El cliente enviará las mismas cookies que el servidor le envió.

4.2.3 El cliente seguiría enviando las mismas cookies, dado que su fecha de expiración es "*Tuesday, 31-Dec-2030 23:12:40 GMT*".

4.2.4 El cliente no enviaría las mismas cookies, dado que la fecha de expiración de las cookies del paquete es "*Tuesday, 31-Dec-2030 23:12:40 GMT*".

4.2.5 Las cookies que envía el servidor al cliente fueron las siguientes: *Cookie: Nif=123456789A; Nombre=Andres.*

4.2.6 El servidor le enviaría estas cookies ya que el cliente se las reenvío.

4.2.7 ***Set-Cookie:*** *Nif=123456789A; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=http://www2.com; Path=/facturas;*
Set-Cookie: *Nif=Andres; Expires=Tuesday, 31-Dec-2030 23:12:40 GMT; Domain=http://www2.com; Path=/facturas;*

4.2.8 El cliente enviaría esas mismas cookies dado que la petición de ese recurso comienza exactamente con ese path de dominio http://www2.com.

4.2.9 El cliente enviaría esas mismas cookies dado que la petición de ese recurso comienza exactamente con ese path de dominio http://www2.com.

4.2.10 El cliente no enviaría esas cookies dado que la petición de ese recurso no comienza exactamente con ese path de dominio http://www2.com.

5. Comunicación a través de un proxy HTTP

5.1 La dirección IP del cliente es **13.0.0.13**.
La dirección IP del proxy es **23.0.0.23**.
La dirección IP del servidor es **22.0.0.22**.

5.2 La diferencia entre la petición HTTP que realiza el cliente y la petición HTTP que realiza el proxy es que el proxy actúa como intermediario entre el cliente y el servidor final y procesa la solicitud en nombre del cliente, es decir, cuando un cliente realiza una solicitud HTTP a través de un proxy, la solicitud se envía primero al proxy, luego reenvía la solicitud al servidor final y, una vez que recibe la respuesta del servidor final, la envía de vuelta al cliente.

5.3 El nombre de la máquina que emplea el servidor es **Apache/2.2.9 (Debian)**.

5.4 Sí, es posible que la petición que realiza el proxy contenga información que indique que el proxy tiene almacenada en su caché la página solicitada.

5.5 El número de conexiones entre cliente y servidor es 2.

5.6 Se está descargando una imagen JPEG.

5.8 Sí se puede saber, ya que si en la cabecera HTTP de los paquetes aparece el campo "***Via***", significa que dicho campo ha sido añadido por un proxy, es decir, que ese paquete ha pasado por x proxies.
Otra opción podría ser comprobar la dirección IP origen y destino de cada mensaje, sabiendo qué dirección IP corresponde al servidor, al cliente y al proxy.

5.9 En el campo "***Via***" se puede encontrar el nombre del proxy: **www3**.

5.10 Sí, al igual que en el apartado 8, se puede observar si el campo "Via" aparece en los paquetes en la cabecera HTML o no.

6. Caches en HTTP

6.1 Caché en un proxy HTTP

6.1.1 La dirección IP del cliente es **12.0.0.100**.
La dirección IP del proxy es **12.0.0.1** y **11.0.0.1**.
La dirección IP del servidor es **14.0.0.100**.

6.1.2 El cliente realiza una petición del recurso **http://14.0.0.100/index.html** al servidor mediante el proxy y una vez obtiene la respuesta, se cierra la conexión, tanto por el lado cliente y el lado servidor con el proxy. El cliente inicia una nueva conexión y solicita nuevamente el mismo recurso que solicitó anteriormente, sin embargo, el proxy al poseer la información que solicita el cliente en su caché, no necesita abrir una nueva conexión con el servidor de nuevo, sino que envía la respuesta en base al contenido de su caché.

6.1.3 Los campos relevantes con respecto al tratamiento de caché que incluyen en las líneas de cabecera del servidor son "***Cache-Control: public, max-age=60***", esto quiere decir que el recurso puede ser almacenado en caches de cualquier tipo y 60 es el número de segundos que el recurso tiene de vigencia; y el "***ETag: "4006-78-490a4f331b500""***, el cual identifica de manera única un recurso para su posterior validación y se emplea para revalidar un recurso caducado.

6.1.4 El cliente realiza una petición del mismo recurso que había solicitado hace unos segundos al proxy.

6.1.5 Se puede saber que el contenido proviene de una caché dado que en el paquete 16 aparecen campos como "***Cache-Control***", "***Expires***" y "***Age***" que confirman que su contenido proviene de una caché.

6.1.6 El cliente puede no poseer caché al no encontrarse ninguno de los campos mencionados en el apartado anterior en sus peticiones.