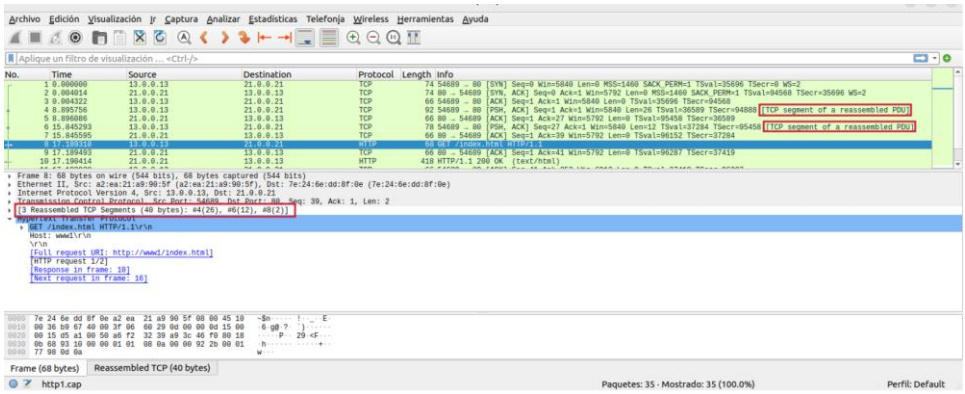
Práctica 5: HTTP

Nota previa (Recordatorio)



Varios reassembles tcp segments formaras más tarde un paquete entero, si pinchas en el paquete entero y despliegas las pestañas veras de que numero de fragmentos está formado.

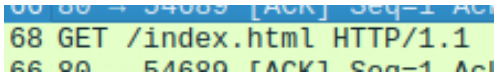
1. Comunicación cliente-servidor HTTP

1. Indica qué dirección IP es la de la máquina cliente HTTP y cuál la del servidor.

El primer paquete dirección 13.0.0.13 - 21.0.0.21 es de client-servidor. Ademas podemos ver el paquete con (SYN).

El segundo paquete va con dirección 21-13. Contiene (SYN, ACK) y pertenece al servidor.

2. Indica qué versión HTTP se utiliza en esta comunicación.



En el paquete 8 podemos ver como especifica la versión 1.1.

3. Indica el número de conexiones que se ven en el fichero de captura, y si los recursos del mismo servidor se transfieren todos por la misma conexión TCP o se usa una conexión TCP diferente para cada uno.

Hay 4 conexiones, dos por parte del cliemte y otras dos por parte del servidor.

Cliente: Paquetes 8 y 16.

Servidor: Paquetes 10 y 30.

- Normalmente una página web está compuesta por varios recursos alojados en diferentes servidores.
- Tras obtenerse el recurso principal, de él se extrae la relación de recursos adicionales que forman parte de la misma página:
  - Para cada nuevo servidor, se abrirá una conexión TCP nueva para pedirle sus recursos.
  - Si hay varios recursos en un mismo servidor, suelen solicitarse todos ellos a través de la misma conexión TCP.

Se encuentran en la misma conexión tcp ya que posteriormente pide una imagen, pero en el mismo servidor, por lo que no hace falta cambiar de conexión tcp.

4. ¿Cuántas peticiones GET observas desde el cliente?

Dos. La primera en el paquete 8 y la segunda en 16. Primero pide la dirección de la página y luego una imagen.

5. ¿Cuántas URLs crees que ha escrito el usuario en el navegador para obtener dicha captura? ¿Cuál/es? ¿Por qué?

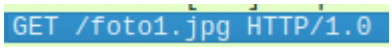
Una única URl, que es la página a la que squería acceder.

6. Fíjate en el contenido de la página index.html que se ha descargado el cliente. ¿Qué crees que ocurrirá cuando el navegador se haya descargado index.html? Abre la captura http2.cap y responde a las siguientes preguntas:

(Contenido de la página index.html)

Se le proporcionará la dirección al cliente y podrá acceder.

7. Indica qué versión HTTP se utiliza en esta comunicación.



Se está usando la versión 1.0.

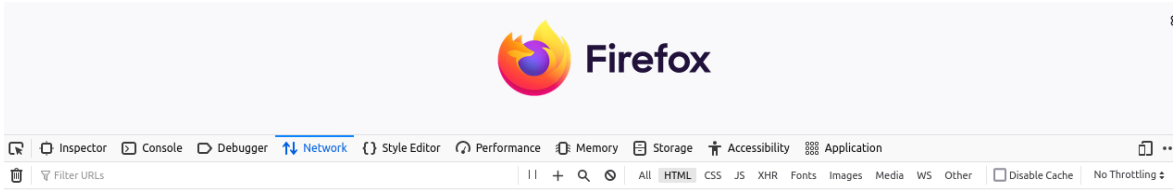
8. Indica el número de conexiones que se ven en el fichero de captura, y si los recursos del mismo servidor se transfieren todos por la misma conexión TCP o se usa una conexión TCP diferente para cada uno

Cliente: 4, 27, 43 y 56.

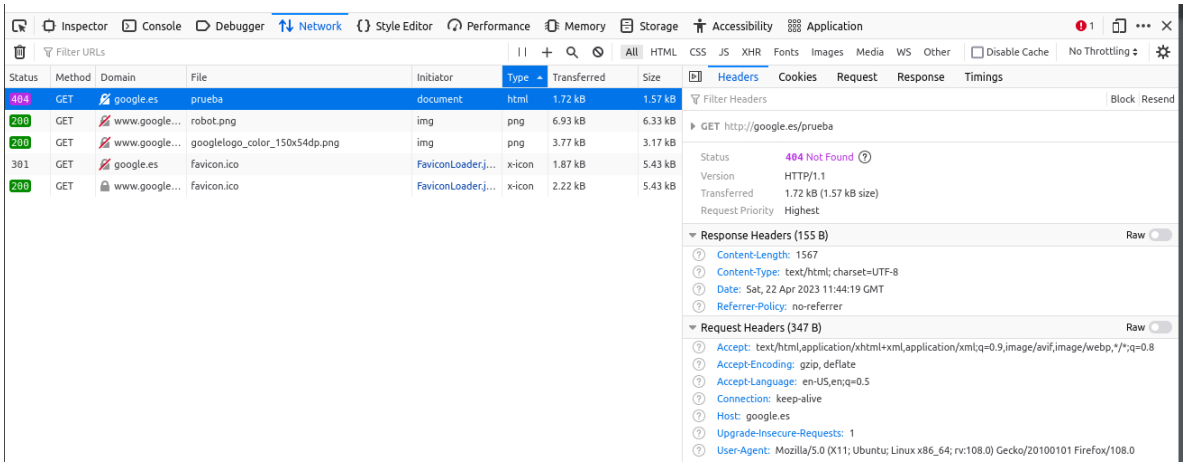
Servidor: 6, 20, 38, 54 y 66.

Es una conexión tcp del mismo servidor ya que el servidor solo está pidiendo fotos y no es necesario la redirección a otro servidor.

2. Diferentes tipos de respuestas de un servidor

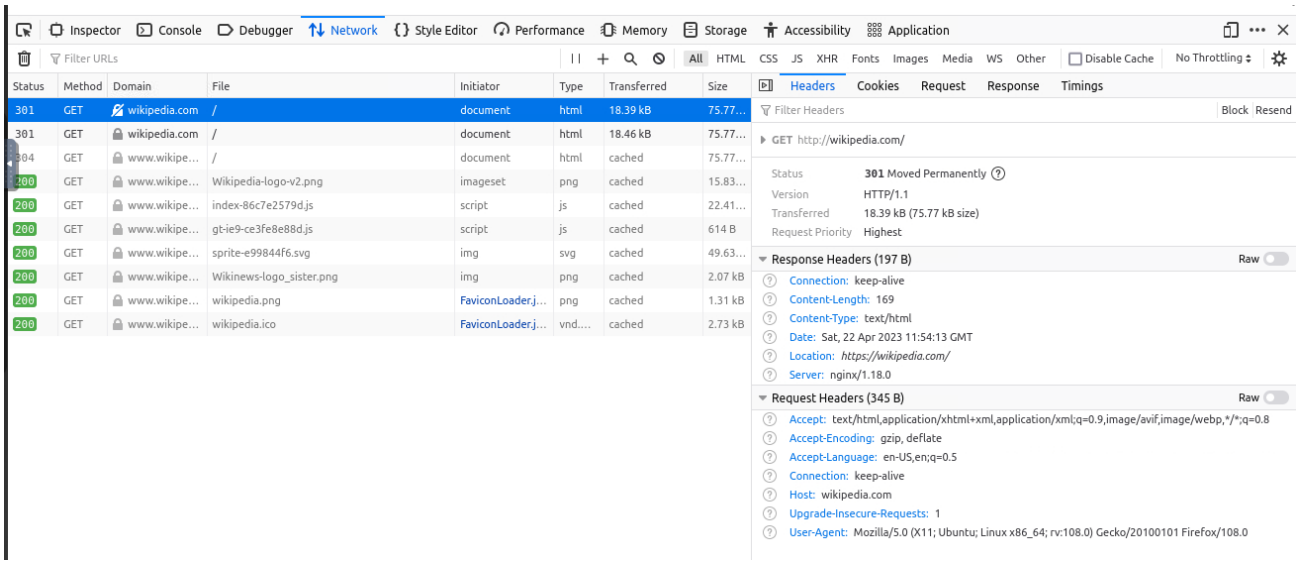


1. Escribe en la barra de dirección la URL http://www.google.es/prueba. Selecciona en el panel inferior la pestaña 'Todos' y pulsa sobre la primera petición GET que aparece (la del fichero 'prueba') y en el panel que se abre a la derecha selecciona la pestaña Cabeceras, como se muestra en la figura 3.

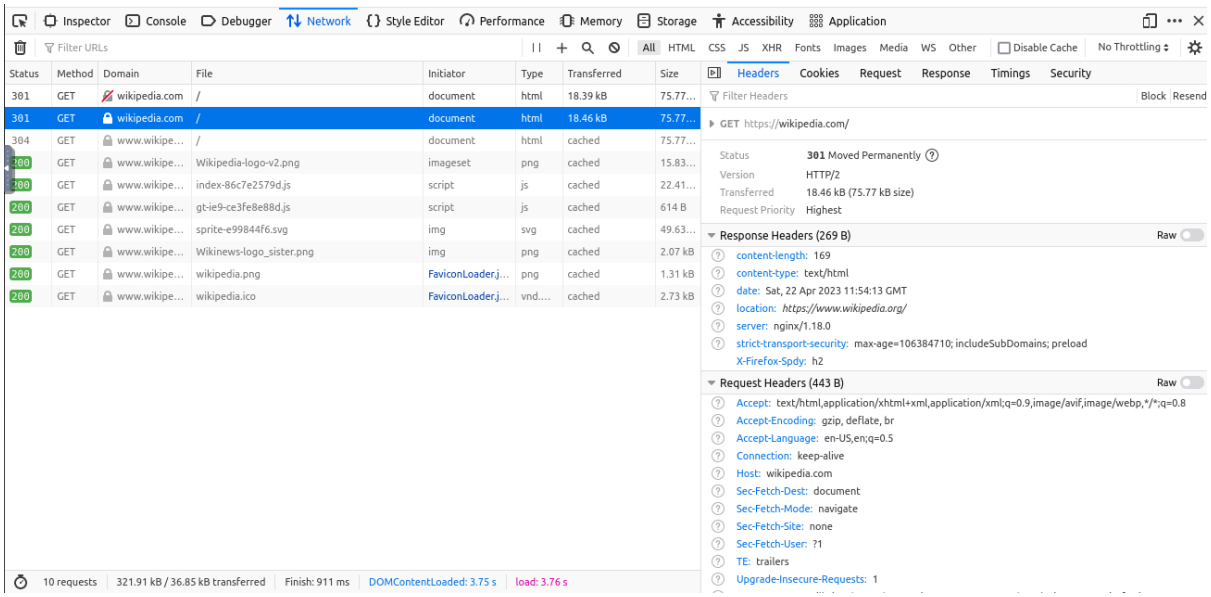


2. Escribe ahora en la barra de dirección la URL http://www.wikipedia.com. En la lista de peticiones, desplázate al principio del todo para localizar el primer GET. Seleccionando ese primer GET mira las líneas de cabecera que ves en la respuesta y explica la segunda petición GET.

Primera solicitud get:



Segunda solicitud get:



3. Formularios en HTTP

1. Indica el número de conexiones entre cliente y servidor que aparecen en la captura.

Cliente: 6 y 17

Servidor: 8 y 21

2. Busca en la captura el paquete donde el servidor le envía al cliente un formulario. Indica los nombres de los campos del formulario que rellenará el usuario.

En el paquete 6 el cliente solicita un formulario y en el 8 el servidor da el ok y lo entrega. En el formulario tendrá que rellenar el nombre, la edad y un número de teléfono.

3. Indica cómo se decide la manera en la que debe enviar el cliente los datos del formulario (GET/POST). ¿Qué método se está usando en este caso?

- GET:
  - Solicita un recurso, la petición no lleva cuerpo. El recurso solicitado se indica en la línea inicial.
  - Ejemplo: solicitar una página web.
- POST:
  - Envía datos a un recurso del servidor. El recurso se indica en la línea inicial y los datos van en el cuerpo de la petición.
  - Ejemplo: enviar los datos que rellena el usuario a través de un formulario web.

- POST:
  - Envía datos al servidor, normalmente los introducidos por el usuario en un formulario.
  - **Los datos van en el cuerpo**.
  - El *path* de la línea inicial (URL) se refiere normalmente al programa que tratará los datos que se envían.
- GET:
  - GET también permite enviar los datos de un formulario. En este caso, **los datos van en el path de la línea inicial (URL), y no hay cuerpo**.
  - El tamaño de los datos subidos con GET está limitado por el tamaño máximo de una URL (255 caracteres), por lo que NO se utiliza GET para subir datos de formularios grandes.

```
466 GET /cgi-bin/prog2.pl?nombre=Ana&edad=25&telefono=123456789 HTTP/1.0
```

Tanto para pedir el formulario como para entregarlo usa el método get

4. Busca en la captura el paquete donde el cliente envía los datos del formulario al servidor y comprueba que se está realizando con el método que has indicado en el apartado anterior.

Usa método get, ya que los datos los introduce en la línea del URL como se puede ver en las capturas anteriores.

5. ¿Cómo se llama el programa del servidor que va a recibir esos datos?

/cgi-bin/prog2.pl (Captura de arriba)

6. ¿Dónde viajan los datos que el cliente le envía al servidor? ¿Cuáles son esos datos?

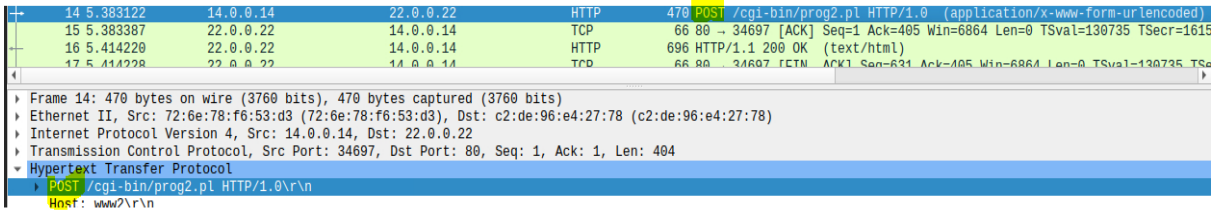
Los datos viajan en el paquete 17 y al ser un mensaje get los datos van en el path de la línea inicial (URL), y no hay cuerpo. Los datos son el nombre, la edad y el número de teléfono.

Abre la captura http4.cap y responde a las siguientes preguntas:

9. Busca en la captura el paquete donde el servidor le envía al cliente un formulario. Indica los nombres de los campos del formulario que rellenará el usuario.

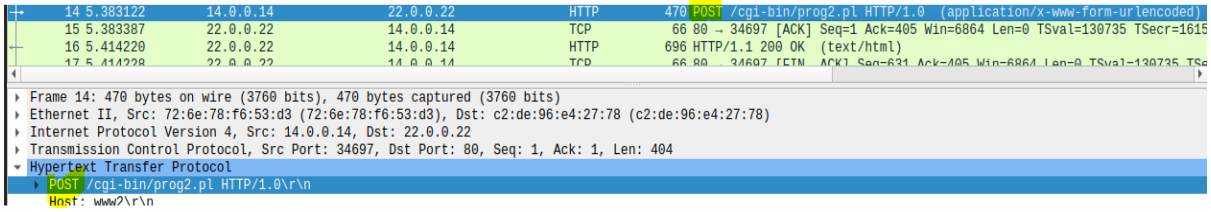
En el fragmento 6, en respuesta a la solicitud del formulario en el fragmento 4.

10. ¿Qué método se usa para enviar los datos al servidor en este caso? ¿Cómo lo sabes?



Se usa el método post.

11. Busca en la captura el paquete donde el cliente le envía los datos del formulario al servidor y comprueba que se está realizando con el método que has indicado en el apartado anterior.



12. ¿Cómo se llama el programa del servidor que va a recibir esos datos?

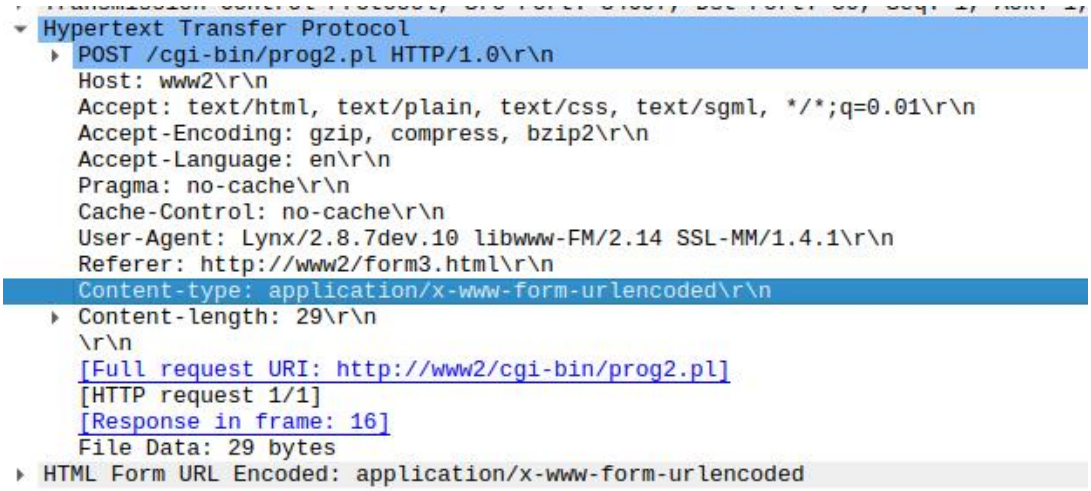
/cgi-bin/prog2.pl

13. Indica en qué parte del mensaje van los datos del formulario que el cliente le envía al servidor.

- POST:
  - Envía datos al servidor, normalmente los introducidos por el usuario en un formulario.
  - **Los datos van en el cuerpo**.
  - El *path* de la línea inicial (URL) se refiere normalmente al programa que tratará los datos que se envían.

Al ser un mensaje post los datos del formulario van en el cuerpo.

14. Indica qué cabecera es la que representa el tipo de contenido que el cliente envía al servidor y cuál es su valor.

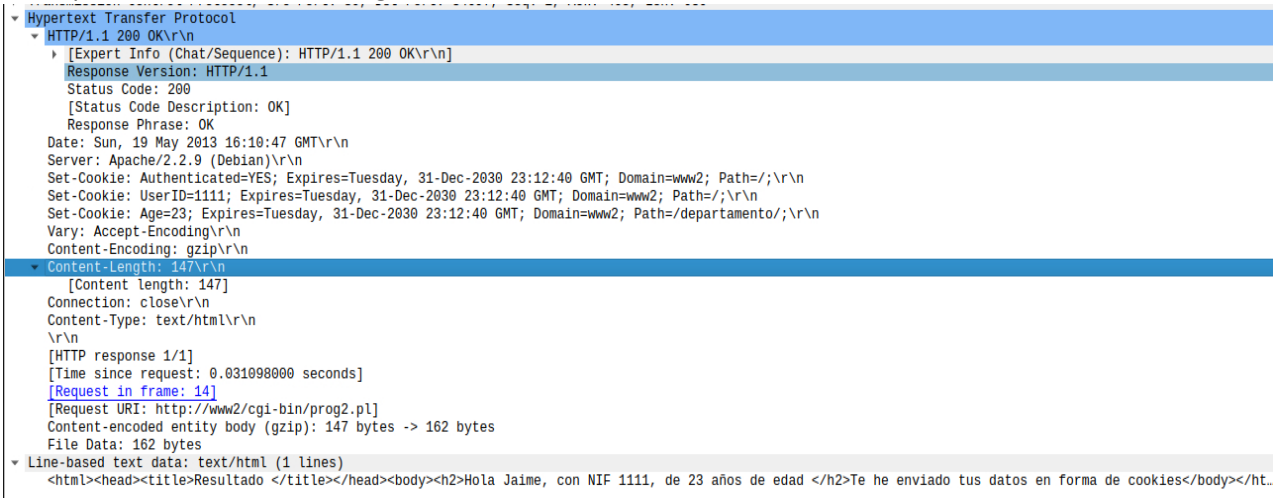


15. Explica si en este caso es necesario la cabecera Content-Length en el mensaje HTTP que el cliente envía al servidor con los datos del formulario. ¿Por qué?

Si es necesario ya que esta cabecera especifica la longitud exacta de esos datos. Sin esta cabecera, el servidor no sabrá cuántos bytes esperar y no podrá procesar el formulario correctamente.

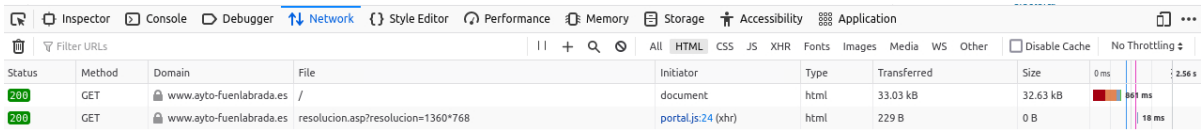
16. Observa si el servidor envía alguna respuesta cuando recibe los datos del formulario del cliente. En caso afirmativo localiza el número de paquete y observa en las cabeceras HTTP: tipo de contenido, longitud y cuerpo del mensaje.

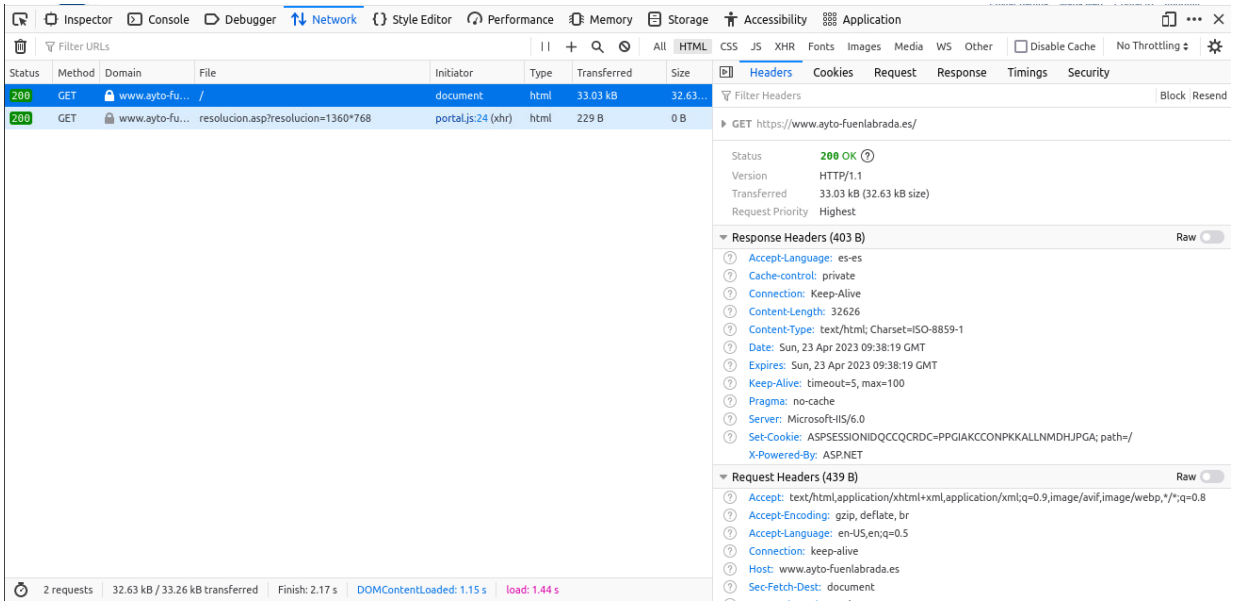
La respuesta se realiza en el fragmento 16.



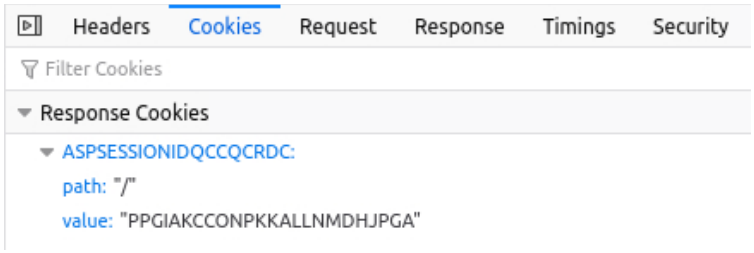
4. Cookies

4.1. Almacén de cookies en el navegador Firefox

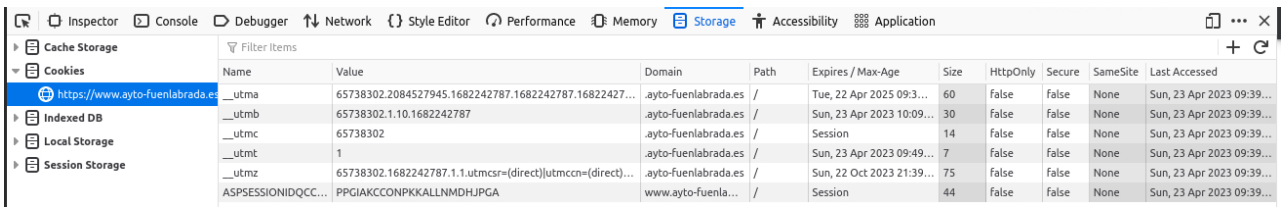




2. Pulsa en el deslizador de 'Sin procesar' en las cabeceras de la respuesta, o selecciona la pestaña "Cookies" para poder ver de forma más clara el contenido de las cookies. Observa que no hay campo 'Domain', lo que indica que la cookie sólo es válida para el servidor que la ha enviado, y que tampoco hay fecha de expiración, lo que indica que es una cookie de sesión, que se eliminará cuando se cierre el navegador.

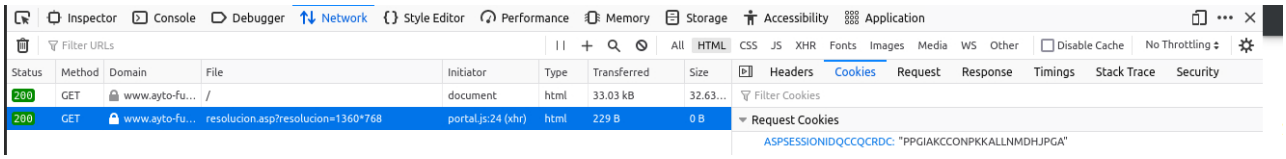


3. Selecciona ahora la herramienta de desarrollador "Almacenamiento" y en el panel de la izquierda despliega "Cookies" para ver las cookies obtenidas al descargar esta página, véase la figura 5. Observa como ahí aparecen rellenados los campos de 'Domain' y 'Expires'.

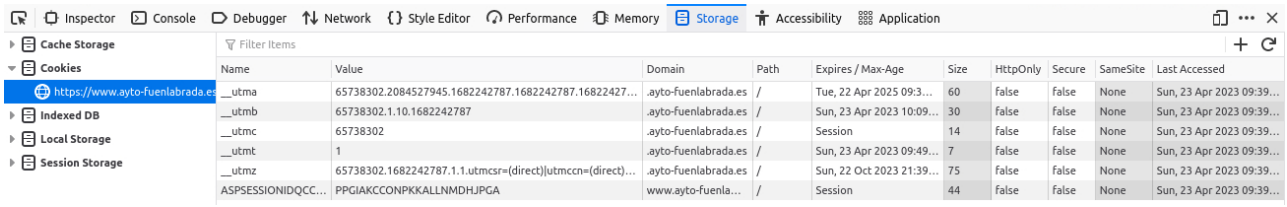


4. Vuelve a la herramienta de desarrollador "Red" y pulsa sobre la segunda petición GET que aparece, y observa las cookies que envía el cliente al servidor. Comprueba como se trata de la

cookie obtenida del servidor en el primer GET. Activa 'Sin procesar' o la pestaña Cookies para poder ver mejor los valores que se envían.



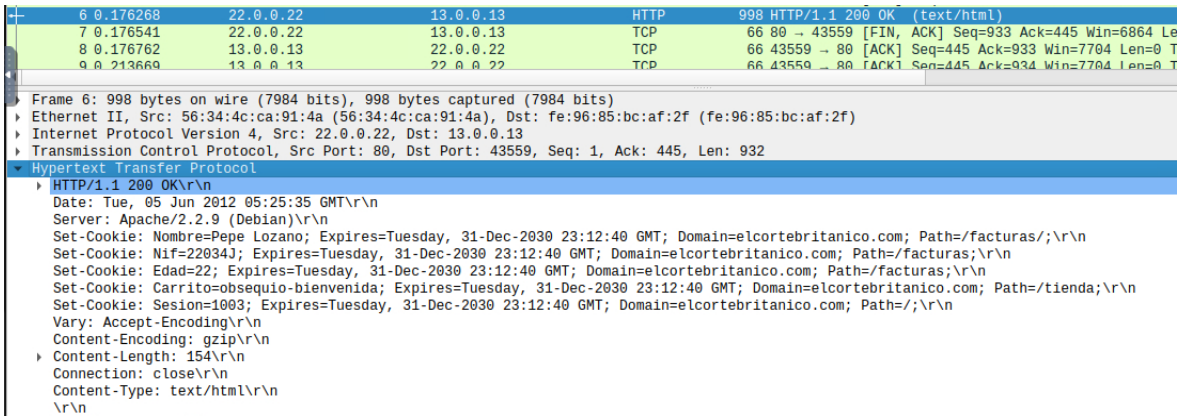
(Mismo número de cookies que el apartado anterior)



4.2. Envío de Cookies en mensajes HTTP

1. Indica qué cookies envía el servidor al cliente:

El servidor le envía 5 cookies, que son:



2. Indica qué cookies enviará el cliente al servidor cuando acceda a la página con la URL: http://elcortebritanico/tienda/index.html

Enviará esta cookie ya que el path de la url es /tienda, al igual que la cookie.

3. ¿Y si el cliente accediera en el año 2025 a dicha URL?

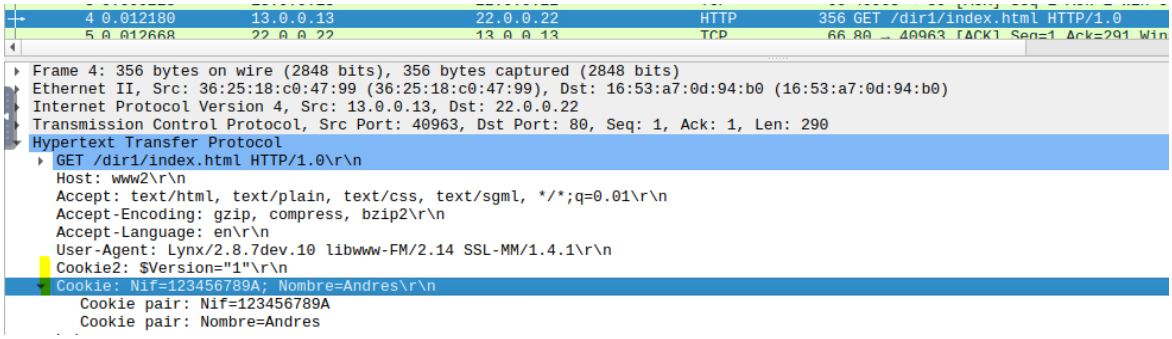
También le valdría ya que la fecha de expiración es en 2030.

4. ¿Y si el cliente accediera en el año 2035 a dicha URL?

La cookie habría expirado y ya no valdría.

Abre la captura http6.cap y responde a las siguientes preguntas:

5. Indica qué cookies el cliente está enviando al servidor.



6. ¿Por qué crees que le envía dichas cookies?

Porque son cookies almacenadas anteriormente y que aún no han expirado. Las envía porque cumplen las condiciones que pide la página.

7. Escribe un ejemplo de las posibles cabeceras que le habrá enviado dicho servidor al cliente previamente.

Pues viendo la cap y viendo que las cookies son el nombre y el nif. Probablemente en el pasado el servidor le pidio una encuesta con estos datos.

La cabecera podría ser:

Set-cookie: Nombre=Andrés; Expires=Tuesday, 28-May-2029 17:55:00 GMT; Domain=encuestastrabajo.com; Path=/socorrista/.

Set-cookie: Nif=123456789A; Expires=Tuesday, 28-May-2029 17:55:00 GMT; Domain=encuestastrabajo.com; Path=/socorrista/.

8. A partir de la información de la captura ¿si el cliente accede a otra página con la URL: http://www2/dir1/dir2/index.html enviaría esas mismas cookies, más cookies o menos?

Enviara estas cookies también porque comienzan con el mismo path, pero al haberlas aceptado ahora no será necesario al cambiar de servidor.

Pero como en el servidor estaremos buscando algo más específico que en la página principal, saldrán nuevas cookies para aceptar.

Ya que si nos metemos en aliexpres por ejemplo, no nos saltan las cookies de ropa, electrodomésticos etc. Cuando inicias te salen unas cookies más generales, y cuando vas especificando que buscas, te saldrán las cookies de eso en específico

9. A partir de la información de la captura ¿crees que si el cliente accede a otra página con la URL: http://www2/index.html enviaría esas mismas cookies, más cookies o menos?
Enviar las cookies que tengan el mismo path que la url anterior. Las cookies con path diferentes no se envían, y luego una vez accedida a la nueva url quizá sean necesarias otras diferentes.

10. A partir de la información de la captura ¿crees que si el cliente accede a otra página con la URL: http://www/index.html enviaría esas mismas cookies, más cookies o menos?

Se enviarán diferentes cookies, más en específico se enviarán las cookies del aparatado anterior, ya que sí que tienen el mismo path.

5. Comunicación a través de un Proxy HTTP

1. Indica qué dirección IP es el cliente, el proxy y el servidor final.

Cliente: 13.0.0.13

Servidor: 22.0.0.22

Proxy: 23.0.0.23

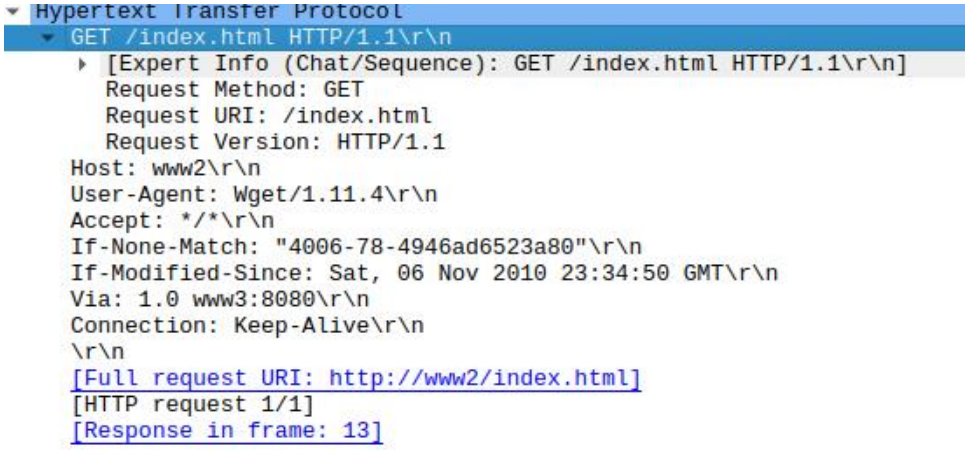
2. ¿Qué diferencia la petición HTTP que realiza el cliente de la petición que realiza el proxy?

El cliente pide directamente la URL de la paginas, mientras que el proxy lo que pide es el path

3. Identifica el nombre de la máquina donde se encuentra el servidor HTTP.

ww2

4. ¿Se puede saber de la petición que realiza el proxy que dicho proxy tiene almacenada en su caché esa página?



Estos son los datos que nos proporciona el proxy, no podemos saber su caché

Abre la captura http8.cap y responde a las siguientes preguntas:

5. Indica el número de conexiones entre cliente y servidor que aparecen en la captura.

Cliente-Servidor: Paquetes 6 y 14

Servidor-Cliente: Paquetes 8 y 34

6. Explica qué es lo que se está descargando el cliente del servidor HTTP y cuantos objetos se descarga.

Se está descargando una imagen. Se descarga 8 objetos.

7. Observa en las cabeceras HTTP el tipo de contenido de cada uno de los objetos.

Los fragmentos del paquete de la foto son tcp, por lo que no tienen http.

8. ¿Podrías saber si los paquetes capturados se corresponden a la comunicación entre un cliente y un proxy HTTP, entre un cliente y servidor final HTTP o entre un proxy y el servidor final HTTP? ¿Por qué?

Los paquetes capturados son ente cliente y proxy, ya que los paquetes del cliente solicitan una http entera y los paquetes enviados por el servidor llevan la cabecera via.

9. Sabiendo que la comunicación se ha realizado a través de un proxy HTTP, mira las cabeceras HTTP que envía dicho proxy para ver si en ellas existe alguna que muestre cuál es su nombre.

Abre la captura http9.cap y responde a las siguientes preguntas:

10. ¿Podrías saber si los paquetes capturados se corresponden a la comunicación entre un cliente y un proxy HTTP, entre un cliente y servidor final HTTP o entre un proxy y el servidor final?

Es una comunicación en la que el cliente pide información sobre una página, pero el proxy ya tiene esa información del servidor por lo que no hace falta volver a preguntarle. Es una comunicación entre cliente y proxy.

6. Cachés en HTTP

6.1. Caché en un proxy HTTP

1. Indica cuáles son las direcciones IP del cliente, proxy y servidor web. ¿Cómo lo sabes?
Cliente: 12.0.0.100

Servidor: 12.0.0.1 / 11.0.0.1

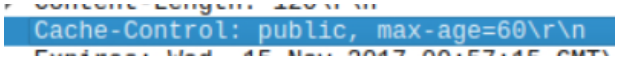
Proxy: 14.0.0.100

Se sabe sacando el camino que sigue la información.

2. Explica qué es lo que ocurre en estas capturas.

Primero el cliente solicita al proxy información sobre el servidor, esto ocurre dos veces. Mas tarde el proxy solicita esta info al servidor y este le responde que no ha cambiado nada. (Supongo que no ha cambiado nada respecto a al info que tenía almacenada el proxy en su caché)

3. Localiza los campos relevantes con respecto al tratamiento de caché que incluye en las líneas de cabecera el servidor. ¿Qué significan?



Significa que se puede guardar en una caché pública y que tiempo máximo de almacena antes de que se convierta en información antigua es de 60 seg.

4. Explica qué ocurre en la segunda consulta que realiza el cliente.

El cliente realiza una consulta y el servidor le responde con la información que tenía guardada en la cache.

5. Viendo los paquetes 14 y 16 de la captura http10.cap indica cómo se puede saber que el contenido proviene de una caché.

Si miramos la respuesta del fragmento 16, vemos que en el age tiene 12/r/ que es mayor que 0, por lo que el contenido estaba guardado en la caché

6. ¿Crees que el cliente tiene caché?

No, porque si no no sería necesaria la respuesta del proxy, ya como el proxy tiene guardada la información en la cache, el cliente también podría haberlo hecho en caso de tener caché.