

PRACTICA 1

SERVICIOS TELEMÁTICOS

Jorge López Saura

Dni: 48745793-F

Correo electronico: jorge.lopez5@um.es

Curso: 3º

Subgrupo: 1.2

ÍNDICE

1.Descripción de la implementación del servicio web.....	3-4
2.Trazas.....	5-10
3.Problemas en el desarrollo del escenario.....	11

1.Descripción de la implementación del servicio web

En primer lugar tras recibir la petición HTTP se procede a leer los datos recibidos en el socket (mediante la función read).

El siguiente paso es parsear (mediante la función strtok) las cabeceras de la petición. En este caso se han tenido en cuenta la línea de solicitud, para realizar la validación, la cabecera “connection” para comprobar si la conexión a de ser persistente o no y la cabecera “Cookie” para comprobar el valor de la cookie.

Para comprobar que una petición HTTP es valida se comprueba que la línea de solicitud contenga el comando “GET” o “POST” , que la ruta introducida es valida (se comprueba mediante el uso de una expresión regular) ,que la versión de HTTP sea la 1.1 (HTTP/1.1) y que haya un espacio entre cada uno de estos tres componentes. En el caso de que al menos una de estas cuatro condiciones sea falsa se devolverá como respuesta un “400 Bad Request” con un fichero HTML indicando que la petición no es valida.

Si la petición es valida se procede a buscar el recurso solicitado en el servidor.

En caso de que el directorio sea “/” se devuelve un mensaje de respuesta “200 OK” con el fichero “index.html”.

En el caso de que se quiera acceder a un directorio del servidor en el cual no se tengan permisos (para ello se comprueba si la ruta tiene la subcadena “..”) se devolverá un mensaje de respuesta “403 Forbidden” con un fichero HTML indicando que no se puede acceder al directorio solicitado.

En el caso de que no se encuentre el recurso solicitado se devolverá un mensaje de respuesta “404 Not Found” con un fichero HTML indicando que no se encuentra el recurso.

Por ultimo, en otro caso se comprobará en primer lugar que la extensión del recurso solicitado esta soportada y en ese caso se devolverá un mensaje de respuesta “200 OK” con el recurso solicitado. En el caso de que la extensión no este soportada se devolverá un mensaje de respuesta “404 Not Found”.

Los mensajes de respuesta HTTP “200 OK” se envían con las siguientes **cabeceras**:

Date: Fecha y hora actual.

Server: Nombre y versión del software del servidor.

Content-length: Longitud en bytes del cuerpo de la respuesta.

Connection: Indica el tipo de conexión.

Content-Type: Tipo MIME que identifica el tipo de dato de la respuesta .

Set-Cookie: El servidor solicita al cliente que almacene una cookie. En este caso esta cabecera tiene dos atributos:

-counter: Indica el numero de accesos al servidor

-Max-Age: Indica el tiempo de vida de la cookie en segundos.

El envío del fichero solicitado se realiza en bloques de máximo 8 kb.

Los mensajes de respuesta “404 Not Found”, “403 Forbidden” y “400 Bad Request” tendrán las mismas cabeceras indicadas anteriormente menos la cabecera “Set-Cookie”.

En cuanto a la persistencia, como se ha mencionado anteriormente se comprueba la cabecera “connection ”de la petición HTTP. Si esta cabecera tiene el valor “keep-alive” la conexión será persistente, en caso de que tenga el valor “close” no será persistente.

La persistencia se ha implementado mediante el uso de la función “select” ,que comprueba si ha habido cambios en algún descriptor de fichero y añade a un conjunto FD_SET los descriptors modificados. En este caso la función “select” solo comprueba un descriptor, el que se pasa como parámetro a la función “process_web_request”, ya que select se invoca dentro de esta función. Si este descriptor es modificado (la función “select” devuelve un valor mayor a cero) se procesara la petición, en caso contrario (la función “select” devuelve 0) debido a que se ha producido un timeout se cerrara la conexión.

En cuanto a las cookies, se utiliza un contador “cookieCounter” para comprobar el numero de accesos al servidor. Esta cookie se crea tras realizar la primera petición, y también se crea en el caso de que la cookie anterior haya expirado. Cuando se crea se le asigna el valor 1 al contador y se establece la fecha de expiración. Si el valor del contador llega a 10 se devolverá un mensaje de respuesta “403 Forbidden” indicando que no se puede acceder al recurso.

2. Trazas

En esta sección vamos a comprobar como se comporta el servidor ante cada uno de los casos mencionados anteriormente, para ello se analizaran las trazas generadas por wireshark para cada uno de estos casos.

En primer lugar vamos a realizar una petición HTTP al servidor solicitando el recurso “index.html”:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.236.3	192.168.236.4	SSH	102	Client: Encrypted packet (len=36)
2	0.001387348	192.168.236.4	192.168.236.3	SSH	182	Server: Encrypted packet (len=116)
3	0.001411283	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=117 Win=1444 Len=0 TSval=1727174776
4	3.100473623	192.168.236.3	192.168.236.4	TCP	74	42334 → 8080 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1727177876
5	3.100725851	192.168.236.4	192.168.236.3	TCP	74	8080 → 42334 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1727177876
6	3.100743661	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1727177876
7	3.100878066	192.168.236.3	192.168.236.4	HTTP	413	GET / HTTP/1.1
8	3.101400840	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [ACK] Seq=1 Ack=348 Win=30080 Len=0 TSval=5122093
9	3.101412038	192.168.236.4	192.168.236.3	TCP	259	8080 → 42334 [PSH, ACK] Seq=1 Ack=348 Win=30080 Len=193 TSval=5122093
10	3.101416915	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=194 Win=30336 Len=0 TSval=172717
11	3.101513821	192.168.236.4	192.168.236.3	SSH	758	Server: Encrypted packet (len=692)
12	3.101523045	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=809 Win=1444 Len=0 TSval=1727177877
13	3.101532091	192.168.236.4	192.168.236.3	HTTP	581	HTTP/1.1 200 OK (text/html)
14	3.101537604	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=709 Win=31360 Len=0 TSval=172717
15	3.233605727	192.168.236.3	192.168.236.4	HTTP	413	GET /index_files/logo-um.jpg HTTP/1.1
16	3.234084954	192.168.236.4	192.168.236.3	TCP	260	8080 → 42334 [PSH, ACK] Seq=709 Ack=695 Win=31104 Len=194 TSval=172717
17	3.234101842	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=903 Win=32512 Len=0 TSval=172717
18	3.234137046	192.168.236.4	192.168.236.3	TCP	8754	8080 → 42334 [ACK] Seq=903 Ack=695 Win=31104 Len=8688 TSval=512
19	3.234146160	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=9591 Win=49792 Len=0 TSval=17271
20	3.234234379	192.168.236.4	192.168.236.3	SSH	798	Server: Encrypted packet (len=732)
21	3.234241193	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1541 Win=1444 Len=0 TSval=172717806
22	3.234251782	192.168.236.4	192.168.236.3	HTTP	936	HTTP/1.1 200 OK (JPEG JFIF image)
23	3.234256143	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=10461 Win=52736 Len=0 TSval=1727
24	13.235767295	192.168.236.3	192.168.236.4	TCP	66	[TCP Keep-Alive] 42334 → 8080 [ACK] Seq=694 Ack=10461 Win=52736
25	13.236300337	192.168.236.4	192.168.236.3	TCP	66	[TCP Keep-Alive ACK] 8080 → 42334 [ACK] Seq=10461 Ack=695 Win=3
26	13.244985055	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [FIN, ACK] Seq=10461 Ack=695 Win=31104 Len=0 TSval=1727
27	13.245010981	192.168.236.4	192.168.236.3	SSH	174	Server: Encrypted packet (len=108)
28	13.245021649	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1649 Win=1444 Len=0 TSval=172718802

El significado de cada mensaje es el siguiente:

4-6: Establecimiento de la conexión TCP por parte del cliente.

7: Mensaje de solicitud HTTP del cliente al servidor.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.236.3	192.168.236.4	SSH	102	Client: Encrypted packet
2	0.001387348	192.168.236.4	192.168.236.3	SSH	182	Server: Encrypted packet
3	0.001411283	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=
4	3.100473623	192.168.236.3	192.168.236.4	TCP	74	42334 → 8080 [SYN] S
5	3.100725851	192.168.236.4	192.168.236.3	TCP	74	8080 → 42334 [SYN, A
6	3.100743661	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] S
7	3.100878066	192.168.236.3	192.168.236.4	HTTP	413	GET / HTTP/1.1
8	3.101400840	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [ACK] S
9	3.101412038	192.168.236.4	192.168.236.3	TCP	259	8080 → 42334 [PSH, A
10	3.101416915	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] S
11	3.101513821	192.168.236.4	192.168.236.3	SSH	758	Server: Encrypted packet
12	3.101523045	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=
13	3.101532091	192.168.236.4	192.168.236.3	HTTP	581	HTTP/1.1 200 OK (text/html)
14	3.101537604	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] S
▶ Frame 7: 413 bytes on wire (3304 bits), 413 bytes captured (3304 bits) on interface 0 ▶ Ethernet II, Src: PcsCompu_08:3c:9a (08:00:27:08:3c:9a), Dst: PcsCompu_34:de:c0 (08:00:27:34:de:c0) ▶ Internet Protocol Version 4, Src: 192.168.236.3, Dst: 192.168.236.4 ▶ Transmission Control Protocol, Src Port: 42334, Dst Port: 8080, Seq: 1, Ack: 1, Len: 347 ▶ Hypertext Transfer Protocol ▶ GET / HTTP/1.1\r\n Host: 192.168.236.4:8080\r\n User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0\r\n Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n Accept-Language: es-ES;q=0.8,en-US;q=0.5,en;q=0.3\r\n Accept-Encoding: gzip, deflate\r\n Connection: keep-alive\r\n Upgrade-Insecure-Requests: 1\r\n \r\n						

Los aspectos mas importantes a destacar del mensaje de solicitud son los siguientes:

- En primer lugar en la linea de solicitud se usa el comando GET,el directorio solicitado es la carpeta raiz del servidor y se usa la version de HTTP 1.1
- En la cabecera “Host” se ubican la dirección IP y puerto del servidor.
- La cabecera “Connection” tiene el valor Keep-Alive, es decir que la conexión sera persistente (mientras no expire el timeout establecido).

No.	Time	Source	Destination	Protocol	Length	Info
7	3.100878066	192.168.236.3	192.168.236.4	HTTP	413	GET / HTTP/1.1
8	3.101400840	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [ACK] Seq=1 Ack=3
9	3.101412038	192.168.236.4	192.168.236.3	TCP	259	8080 → 42334 [PSH, ACK] Seq=1
10	3.101416915	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=
11	3.101513821	192.168.236.4	192.168.236.3	SSH	758	Server: Encrypted packet (len=)
12	3.101523045	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=80
13	3.101532091	192.168.236.4	192.168.236.3	HTTP	581	HTTP/1.1 200 OK (text/html)
14	3.101537604	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=
15	3.233605727	192.168.236.3	192.168.236.4	HTTP	413	GET /index_files/logo-um.jpg H
16	3.234084954	192.168.236.4	192.168.236.3	TCP	260	8080 → 42334 [PSH, ACK] Seq=70
17	3.234101842	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=
18	3.234137046	192.168.236.4	192.168.236.3	TCP	8754	8080 → 42334 [ACK] Seq=903 Ack=
19	3.234146160	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=
20	3.234224270	192.168.236.4	192.168.236.3	SSH	708	Server: Encrypted packet (len=)
▶ Frame 13: 581 bytes on wire (4648 bits), 581 bytes captured (4648 bits) on interface 0 ▶ Ethernet II, Src: PcsCompu_34:de:c0 (08:00:27:34:de:c0), Dst: PcsCompu_08:3c:9a (08:00:27:08:3c:9a) ▶ Internet Protocol Version 4, Src: 192.168.236.4, Dst: 192.168.236.3 ▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 42334, Seq: 194, Ack: 348, Len: 515 ▶ [2 Reassembled TCP Segments (708 bytes): #9(193), #13(515)] ▶ Hypertext Transfer Protocol ▶ HTTP/1.1 200 OK\r\n Date: Sun Mar 24 16:49:02 2019\r\n Server: UbuntuServer/16.04\r\n Content-Length: 515\r\n Connection: keep-alive\r\n Content-Type: text/html\r\n Set-Cookie: counter=1; Max-Age=120; Path=/\r\n \r\n						

El mensaje numero 13 es el mensaje de respuesta del servidor al cliente. En este caso es una respuesta 200 OK (la petición es valida y el recurso esta disponible).

Como se puede observar se crea una cookie mediante la cabecera “Set-Cookie”, cuyo valor es 1 (counter=1) y su tiempo de vida máximo es de 2 minutos (Max-Age=120).

El siguiente mensaje (numero 15) es otra petición del cliente al servidor, en este caso solicita la imagen contenida en el “index.html” solicitado en la anterior petición.

No.	Time	Source	Destination	Protocol	Length	Info
7	3.100878066	192.168.236.3	192.168.236.4	HTTP	413	GET / HTTP/1.1
8	3.101400840	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [ACK] Seq=1 Ack=348 Win=300
9	3.101412038	192.168.236.4	192.168.236.3	TCP	259	8080 → 42334 [PSH, ACK] Seq=1 Ack=348 Win=300
10	3.101416915	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=194 Win=300
11	3.101513821	192.168.236.4	192.168.236.3	SSH	758	Server: Encrypted packet (len=692)
12	3.101523045	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=809 Win=1444
13	3.101532091	192.168.236.4	192.168.236.3	HTTP	581	HTTP/1.1 200 OK (text/html)
14	3.101537604	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=348 Ack=709 Win=300
15	3.233605727	192.168.236.3	192.168.236.4	HTTP	413	GET /index_files/logo-um.jpg HTTP/1.1
16	3.234084954	192.168.236.4	192.168.236.3	TCP	260	8080 → 42334 [PSH, ACK] Seq=709 Ack=695 Win=300
17	3.234101842	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=903 Win=300
18	3.234137046	192.168.236.4	192.168.236.3	TCP	8754	8080 → 42334 [ACK] Seq=903 Ack=695 Win=300
19	3.234146160	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=9591 Win=300
20	3.234224270	192.168.236.4	192.168.236.3	SSH	708	Server: Encrypted packet (len=722)

▶ Frame 15: 413 bytes on wire (3304 bits), 413 bytes captured (3304 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_08:3c:9a (08:00:27:08:3c:9a), Dst: PcsCompu_34:de:c0 (08:00:27:34:de:c0)
 ▶ Internet Protocol Version 4, Src: 192.168.236.3, Dst: 192.168.236.4
 ▶ Transmission Control Protocol, Src Port: 42334, Dst Port: 8080, Seq: 348, Ack: 709, Len: 347
 ▶ Hypertext Transfer Protocol
 ▶ GET /index_files/logo-um.jpg HTTP/1.1\r\n
 Host: 192.168.236.4:8080\r\n
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0\r\n
 Accept: image/webp,*/*\r\n
 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\n
 Accept-Encoding: gzip, deflate\r\n
 Referer: http://192.168.236.4:8080/\r\n
 Connection: keep-alive\r\n
 Cookie: counter=1\r\n\r\n

Como se puede observar, en este caso, en la linea de solicitud el recurso solicitado es la imagen “logo-um.jpg”.Ademas el mensaje contiene otras cabeceras como “Referer” que contiene la url del documento desde el que se accedió al actual y la cabecera “Cookie” que contiene el valor de la cookie devuelta en el mensaje de respuesta anterior.

No.	Time	Source	Destination	Protocol	Length	Info
16	3.234084954	192.168.236.4	192.168.236.3	TCP	260	8080 → 42334 [PSH, ACK] Seq=709 Ack=695 Win=31104 Len=0 TSval=172
17	3.234101842	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=903 Win=32512 Len=0 TSval=172
18	3.234137046	192.168.236.4	192.168.236.3	TCP	8754	8080 → 42334 [ACK] Seq=903 Ack=695 Win=31104 Len=8688 TSval=172
19	3.234146160	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=9591 Win=49792 Len=0 TSval=172
20	3.234234379	192.168.236.4	192.168.236.3	SSH	798	Server: Encrypted packet (len=732)
21	3.234241193	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1541 Win=1444 Len=0 TSval=172
22	3.234251782	192.168.236.4	192.168.236.3	HTTP	936	HTTP/1.1 200 OK (JPEG JFIF image)
23	3.234256143	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=10461 Win=52736 Len=0 TSval=172
24	13.235767295	192.168.236.3	192.168.236.4	TCP	66	[TCP Keep-Alive] 42334 → 8080 [ACK] Seq=694 Ack=10461 Win=0
25	13.236300337	192.168.236.4	192.168.236.3	TCP	66	[TCP Keep-Alive ACK] 8080 → 42334 [ACK] Seq=10461 Ack=695
26	13.244985055	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [FIN, ACK] Seq=10461 Ack=695 Win=31104 Len=0
27	13.245010981	192.168.236.4	192.168.236.3	SSH	174	Server: Encrypted packet (len=108)
28	13.245021649	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1649 Win=1444 Len=0 TSval=172
29	13.245274702	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [FIN, ACK] Seq=695 Ack=10461 Win=52736 Len=0

▶ Frame 22: 936 bytes on wire (7488 bits), 936 bytes captured (7488 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_34:de:c0 (08:00:27:34:de:c0), Dst: PcsCompu_08:3c:9a (08:00:27:08:3c:9a)
 ▶ Internet Protocol Version 4, Src: 192.168.236.4, Dst: 192.168.236.3
 ▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 42334, Seq: 9591, Ack: 695, Len: 870
 ▶ [3 Reassembled TCP Segments (9752 bytes): #16(194), #18(8688), #22(870)]
 ▶ Hypertext Transfer Protocol
 ▶ HTTP/1.1 200 OK\r\n
 Date: Sun Mar 24 16:49:02 2019\r\n
 Server: UbuntuServer/16.04\r\n
 Content-Length: 9558\r\n
 Connection: keep-alive\r\n
 Content-Type: image/jpeg\r\n
 Set-Cookie: counter=2; Max-Age=120; Path=/\r\n\r\n

La respuesta enviada por el servidor a la petición anterior es el mensaje número 22. En este caso la respuesta no identifica ningún error (respuesta 200 OK) y como se puede observar en este caso se establece una cookie con valor 2 (counter=2) ya que se han realizado 2 peticiones al servidor.

En los mensajes 24,25 se comprueba si la conexión entre cliente y servidor sigue activa.

No.	Time	Source	Destination	Protocol	Length	Info
21	3.234241193	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1541 Win=1444 Len=0 TSval=1727178009 TS
22	3.234251782	192.168.236.4	192.168.236.3	HTTP	936	HTTP/1.1 200 OK (JPEG JFIF image)
23	3.234256143	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [ACK] Seq=695 Ack=10461 Win=52736 Len=0 TSval=17271780
24	13.235767295	192.168.236.3	192.168.236.4	TCP	66	[TCP Keep-Alive] 42334 → 8080 [ACK] Seq=694 Ack=10461 Win=52736 Len
25	13.236300337	192.168.236.4	192.168.236.3	TCP	66	[TCP Keep-Alive ACK] 8080 → 42334 [ACK] Seq=10461 Ack=695 Win=31104
26	13.244985055	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [FIN, ACK] Seq=10461 Ack=695 Win=31104 Len=0 TSval=512
27	13.245010981	192.168.236.4	192.168.236.3	SSH	174	Server: Encrypted packet (len=108)
28	13.245021649	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=1649 Win=1444 Len=0 TSval=1727188020 TS
29	13.245274793	192.168.236.3	192.168.236.4	TCP	66	42334 → 8080 [FIN, ACK] Seq=695 Ack=10462 Win=52736 Len=0 TSval=172
30	13.245551653	192.168.236.4	192.168.236.3	TCP	66	8080 → 42334 [ACK] Seq=10462 Ack=696 Win=31104 Len=0 TSval=5124629
31	23.181872163	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol

En caso de que el timeout expire se cierra la conexión entre cliente y servidor (mensajes 26-30).

En el caso de que se vuelva a realizar una nueva petición tras pasar mas de dos minutos tras la ultima petición (la cookie expira) el valor de la cookie se resetea a 1

No.	Time	Source	Destination	Protocol	Length	Info
688	2636.0804012...	192.168.236.1	192.168.236.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
689	2636.0815497...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
690	2636.0821804...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
691	2644.8445276...	192.168.236.3	192.168.236.4	TCP	74	42346 → 8080 [SYN] Seq=0 Win=29200
692	2644.8448354...	192.168.236.4	192.168.236.3	TCP	74	8080 → 42346 [SYN, ACK] Seq=0 Ack=1
693	2644.8448632...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=1 Ack=1 Win=
694	2644.8453489...	192.168.236.3	192.168.236.4	HTTP	424	GET /logo-um.jpg HTTP/1.1
695	2644.8456035...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42346 [ACK] Seq=1 Ack=359 Wi
696	2644.8461760...	192.168.236.4	192.168.236.3	TCP	260	8080 → 42346 [PSH, ACK] Seq=1 Ack=3
697	2644.8461893...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=195
698	2644.8462440...	192.168.236.4	192.168.236.3	TCP	7306	8080 → 42346 [ACK] Seq=195 Ack=359
699	2644.8462594...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=7435
700	2644.8464285...	192.168.236.4	192.168.236.3	TCP	1514	8080 → 42346 [ACK] Seq=7435 Ack=359
701	2644.8464395...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=8883
702	2644.8464546...	192.168.236.4	192.168.236.3	HTTP	936	HTTP/1.1 200 OK (JPEG JFIF image)
703	2644.8464617...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=9753
704	2644.8466698...	192.168.236.4	192.168.236.3	SSH	798	Server: Encrypted packet (len=732)
705	2644.8466852...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=2381 Wi

▶ [Timestamps]
TCP payload (358 bytes)
▼ Hypertext Transfer Protocol
▶ GET /logo-um.jpg HTTP/1.1\r\n
Host: 192.168.236.4:8080\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n

Como se puede observar en la imagen anterior el cliente además vuelve a solicitar una nueva conexión al servidor ya que la anterior expiró (lineas 691 a 693) y en el mensaje de solicitud (linea 694) no hay ninguna cookie establecida.

No.	Time	Source	Destination	Protocol	Length	Info
691	2644.8445276...	192.168.236.3	192.168.236.4	TCP	74	42346 → 8080 [SYN] Seq=0 Win=29200
692	2644.8448354...	192.168.236.4	192.168.236.3	TCP	74	8080 → 42346 [SYN, ACK] Seq=0 Ack=1
693	2644.8448632...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=1 Ack=1 Win=
694	2644.8453489...	192.168.236.3	192.168.236.4	HTTP	424	GET /logo-um.jpg HTTP/1.1
695	2644.8456035...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42346 [ACK] Seq=1 Ack=359 Wi
696	2644.8461760...	192.168.236.4	192.168.236.3	TCP	260	8080 → 42346 [PSH, ACK] Seq=1 Ack=3
697	2644.8461893...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=195
698	2644.8462440...	192.168.236.4	192.168.236.3	TCP	7306	8080 → 42346 [ACK] Seq=195 Ack=359
699	2644.8462594...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=7435
700	2644.8464285...	192.168.236.4	192.168.236.3	TCP	1514	8080 → 42346 [ACK] Seq=7435 Ack=359
701	2644.8464395...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=8883
702	2644.8464546...	192.168.236.4	192.168.236.3	HTTP	936	HTTP/1.1 200 OK (JPEG JFIF image)
703	2644.8464617...	192.168.236.3	192.168.236.4	TCP	66	42346 → 8080 [ACK] Seq=359 Ack=9753
704	2644.8466698...	192.168.236.4	192.168.236.3	SSH	798	Server: Encrypted packet (len=732)
705	2644.8466852...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=2381 Wi
706	2649.8481967...	PcsCompu_34:de:c0	PcsCompu_08:3c:9a	ARP	60	Who has 192.168.236.3? Tell 192.168
707	2649.8482200...	PcsCompu_08:3c:9a	PcsCompu_34:de:c0	ARP	42	192.168.236.3 is at 08:00:27:08:3c:
708	2649.9872459...	PcsCompu_08:3c:9a	PcsCompu_34:de:c0	ARP	42	Who has 192.168.236.4? Tell 192.168
709	2649.9875527...	PcsCompu_34:de:c0	PcsCompu_08:3c:9a	ARP	60	192.168.236.4 is at 08:00:27:08:3c:

[Timestamps]
 TCP payload (870 bytes)
 TCP segment data (870 bytes)
 [4 Reassembled TCP Segments (9752 bytes): #696(194), #698(7240), #700(1448), #702(870)]
Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
 Date: Sun Mar 24 17:33:04 2019\r\n
 Server: UbuntuServer/16.04\r\n
 Content-Length: 9558\r\n
 Connection: keep-alive\r\n
 Content-Type: image/jpeg\r\n
 Set-Cookie: counter=1; Max-Age=120; Path=/\r\n\r\n

En el mensaje de respuesta (línea 702) se vuelve a establecer el valor de la cookie a 1.

En el caso de que se solicite un recurso que no está en el servidor, éste devuelve un mensaje 404 Not Found como se indica a continuación:

No.	Time	Source	Destination	Protocol	Length	Info
1135	4348.7817459...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1136	4348.7825136...	192.168.236.1	192.168.236.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1137	4348.7829961...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1138	4348.7830082...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1139	4350.5299481...	192.168.236.3	192.168.236.4	TCP	74	42354 → 8080 [SYN] Seq=0 Win=29200 Len=0
1140	4350.5301949...	192.168.236.4	192.168.236.3	TCP	74	8080 → 42354 [SYN, ACK] Seq=0 Ack=1 Win=
1141	4350.5302145...	192.168.236.3	192.168.236.4	TCP	66	42354 → 8080 [ACK] Seq=1 Ack=1 Win=29312
1142	4350.5304306...	192.168.236.3	192.168.236.4	HTTP	423	GET /noesta.gif HTTP/1.1
1143	4350.5307808...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42354 [ACK] Seq=1 Ack=358 Win=300
1144	4350.5311746...	192.168.236.4	192.168.236.3	TCP	222	8080 → 42354 [PSH, ACK] Seq=1 Ack=358 Wi
1145	4350.5311946...	192.168.236.3	192.168.236.4	TCP	66	42354 → 8080 [ACK] Seq=358 Ack=157 Win=3
1146	4350.5312164...	192.168.236.4	192.168.236.3	SSH	798	Server: Encrypted packet (len=732)
1147	4350.5312289...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=3221 Win=144
1148	4350.5313304...	192.168.236.4	192.168.236.3	HTTP	248	HTTP/1.1 404 Not Found (text/html)
1149	4350.5313425...	192.168.236.3	192.168.236.4	TCP	66	42354 → 8080 [ACK] Seq=358 Ack=339 Win=3
1150	4355.5431842...	PcsCompu_34:de:c0	PcsCompu_08:3c:9a	ARP	60	Who has 192.168.236.3? Tell 192.168.236.
1151	4355.5431981...	PcsCompu_08:3c:9a	PcsCompu_34:de:c0	ARP	42	192.168.236.3 is at 08:00:27:08:3c:9a
1152	4355.7153182...	PcsCompu_08:3c:9a	PcsCompu_34:de:c0	ARP	42	Who has 192.168.236.4? Tell 192.168.236.
1153	4355.7157204...	PcsCompu_34:de:c0	PcsCompu_08:3c:9a	ARP	60	192.168.236.4 is at 08:00:27:08:3c:9a

[Timestamps]
 TCP payload (182 bytes)
 TCP segment data (182 bytes)
 [2 Reassembled TCP Segments (338 bytes): #1144(156), #1148(182)]
Hypertext Transfer Protocol
 HTTP/1.1 404 Not Found\r\n
 Date: Sun Mar 24 18:01:30 2019\r\n
 Server: UbuntuServer/16.04\r\n
 Content-Length: 182\r\n
 Connection: keep-alive\r\n
 Content-Type: text/html\r\n\r\n

En el caso de que se realice una petición mal formada (línea 1278) se devolverá un mensaje del tipo 400 Bad Request (línea 1284) como se muestra en la siguiente imagen:

No.	Time	Source	Destination	Protocol	Length	Info
1270	4799.4432579...	192.168.236.1	192.168.236.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1271	4799.4437201...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1272	4799.4437707...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
1273	4799.7859789...	192.168.236.3	192.168.236.4	TCP	74	42358 → 8080 [SYN] Seq=0 Win=29200 Len=0 M
1274	4799.7861641...	192.168.236.4	192.168.236.3	TCP	74	8080 → 42358 [SYN, ACK] Seq=0 Ack=1 Win=28
1275	4799.7861642...	192.168.236.3	192.168.236.4	TCP	66	42358 → 8080 [ACK] Seq=1 Ack=1 Win=29312 L
1276	4802.6739219...	192.168.236.3	192.168.236.4	TCP	81	42358 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=29
1277	4802.6741930...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42358 [ACK] Seq=1 Ack=16 Win=29056
1278	4802.6742082...	192.168.236.3	192.168.236.4	TCP	378	42358 → 8080 [PSH, ACK] Seq=16 Ack=1 Win=2
1279	4802.6750106...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42358 [ACK] Seq=1 Ack=328 Win=30080
1280	4802.6752563...	192.168.236.4	192.168.236.3	TCP	224	8080 → 42358 [PSH, ACK] Seq=1 Ack=328 Win=
1281	4802.6752654...	192.168.236.3	192.168.236.4	TCP	66	42358 → 8080 [ACK] Seq=328 Ack=159 Win=303
1282	4802.6752799...	192.168.236.4	192.168.236.3	SSH	686	Server: Encrypted packet (len=620)
1283	4802.6752853...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=3949 Win=1444
1284	4802.6754360...	192.168.236.4	192.168.236.3	HTTP	285	HTTP/1.1 400 Bad Request (text/html)
1285	4802.6754436...	192.168.236.3	192.168.236.4	TCP	66	42358 → 8080 [ACK] Seq=328 Ack=378 Win=313
1286	4812.6799695...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42358 [FIN, ACK] Seq=378 Ack=328 Wi
1287	4812.6805519...	192.168.236.4	192.168.236.3	SSH	174	Server: Encrypted packet (len=108)
1288	4812.6805707...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=37 Ack=4057 Win=1444
<ul style="list-style-type: none"> ▶ [Timestamps] <ul style="list-style-type: none"> TCP payload (219 bytes) TCP segment data (219 bytes) ▶ [2 Reassembled TCP Segments (377 bytes): #1280(158), #1284(219)] ▼ Hypertext Transfer Protocol <ul style="list-style-type: none"> ▶ HTTP/1.1 400 Bad Request\r\n Date: Sun Mar 24 18:09:02 2019\r\n Server: UbuntuServer/16.04\r\n Content-Length: 219\r\n Connection: keep-alive\r\n Content-Type: text/html\r\n \r\n 						

Por ultimo en el caso de que se realice una petición (línea 2436) para obtener un recurso, al cual no tenemos los permisos requeridos para acceder, el servidor devuelve un mensaje del tipo 403 Forbidden (línea 2442) como se muestra a continuación:

No.	Time	Source	Destination	Protocol	Length	Info
2434	6385.2863637...	192.168.236.3	192.168.236.4	TCP	104	42392 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2
2435	6385.2866370...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42392 [ACK] Seq=1 Ack=39 Win=29056
2436	6385.2866542...	192.168.236.3	192.168.236.4	HTTP	378	GET ../../etc/adduser.conf HTTP/1.1
2437	6385.2868196...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42392 [ACK] Seq=1 Ack=351 Win=3008
2438	6385.2871828...	192.168.236.4	192.168.236.3	TCP	222	8080 → 42392 [PSH, ACK] Seq=1 Ack=351 Win=3008
2439	6385.2871951...	192.168.236.3	192.168.236.4	TCP	66	42392 → 8080 [ACK] Seq=351 Ack=157 Win=3008
2440	6385.2873820...	192.168.236.4	192.168.236.3	SSH	830	Server: Encrypted packet (len=764)
2441	6385.2873918...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=1261 Ack=26337 Win=1
2442	6385.2874052...	192.168.236.4	192.168.236.3	HTTP	248	HTTP/1.1 403 Forbidden (text/html)
2443	6385.2874093...	192.168.236.3	192.168.236.4	TCP	66	42392 → 8080 [ACK] Seq=351 Ack=339 Win=3008
2444	6391.7880220...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2445	6391.7932325...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2446	6391.7932842...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2447	6391.7936354...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2448	6391.7949278...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2449	6391.7949406...	192.168.236.1	255.255.255.255	DB-LSP...	244	Dropbox LAN sync Discovery Protocol
2450	6395.2987017...	192.168.236.4	192.168.236.3	TCP	66	8080 → 42392 [FIN, ACK] Seq=339 Ack=351 W
2451	6395.2990655...	192.168.236.4	192.168.236.3	SSH	174	Server: Encrypted packet (len=108)
2452	6395.2990956...	192.168.236.3	192.168.236.4	TCP	66	37498 → 22 [ACK] Seq=1261 Ack=26445 Win=1
<ul style="list-style-type: none"> TCP segment data (182 bytes) ▶ [2 Reassembled TCP Segments (338 bytes): #2438(156), #2442(182)] ▼ Hypertext Transfer Protocol <ul style="list-style-type: none"> ▶ HTTP/1.1 403 Forbidden\r\n Date: Sun Mar 24 18:35:24 2019\r\n Server: UbuntuServer/16.04\r\n Content-Length: 182\r\n Connection: keep-alive\r\n Content-Type: text/html\r\n \r\n 						

3.PROBLEMAS EN EL DESARROLLO DEL ESCENARIO

En mi caso, no he encontrado ningún problema en el proceso de desarrollo del escenario.