

Session Persistence with Cookies in HTTP

This report investigates the utilization of HTTP cookies in preserving session state between a client and a server. By employing Wireshark, we recorded network traffic throughout a website login session and scrutinized HTTP packets to discern and comprehend the function of cookies.

1. Capturing Network Traffic in Wireshark

Launched Wireshark and chose the active network interface for monitoring.

Initiated packet capture while logging into a test website.

Concluded the traffic capture once the login process was finalized.

2. Filtering for HTTP Traffic

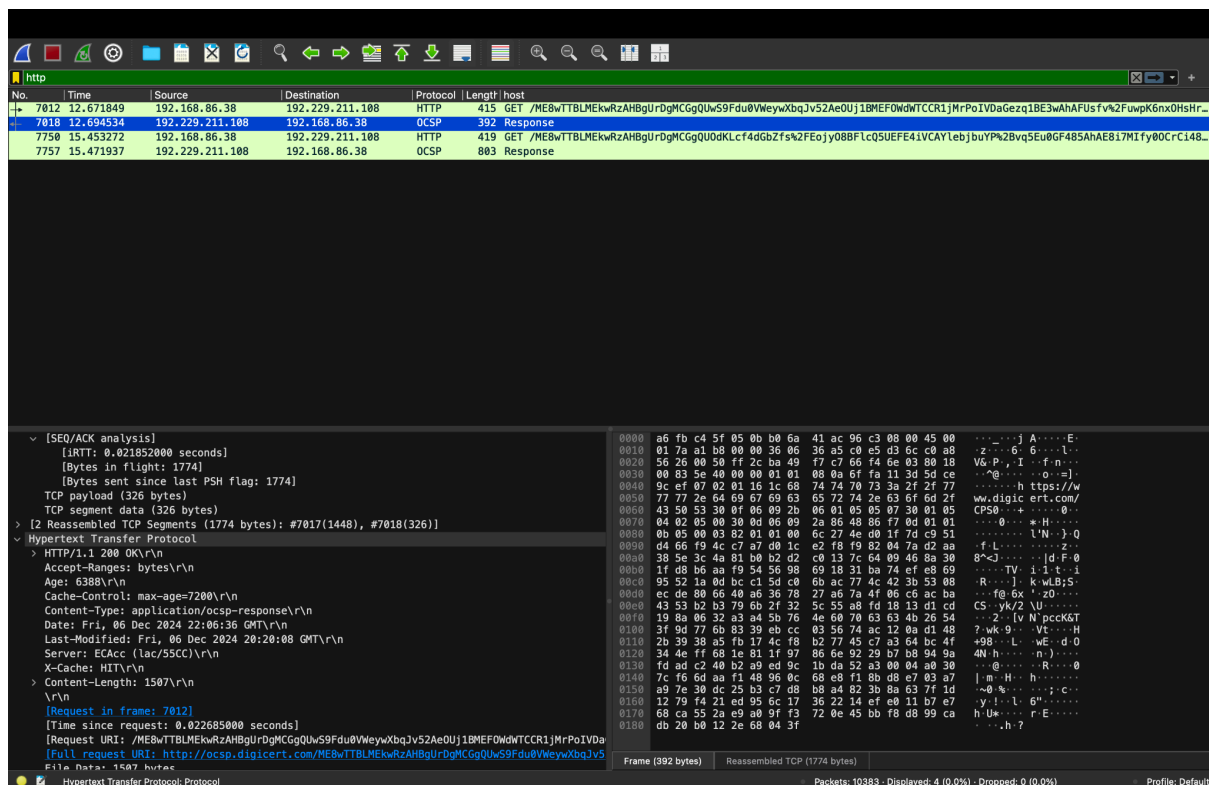
Utilized the http filter in Wireshark to extract HTTP packets from the collected data.

3. Identifying Important Packets

Recognized:

HTTP response packets that included the Set-Cookie header sent by the server.

HTTP request packets that contained the Cookie header from the client.



This screenshot contains HTTP cookies in HTTP/1.1 200 OK/r/n

Cookies and Session cookie

Objective:

Cookies allow the server to recognize the client during multiple requests. In the absence of cookies, the stateless nature of HTTP prevents the maintenance of session continuity.

Process:

The server transmits a Set-Cookie header containing a unique session identifier. The client retains the cookie and incorporates it into future Cookie headers. The server utilizes the session identifier to access data specific to the session.

The image shows a Wireshark packet capture of an HTTP transaction. The packet list on the left shows three packets: a GET request (No. 7012), a 200 OK response (No. 7018), and another GET request (No. 7750). The selected packet (No. 7018) is an HTTP response from 192.229.211.108 to 192.168.86.38. The packet details pane on the left shows the structure of the response, including the status line (200 OK), headers (Accept-Ranges, Age, Cache-Control, Content-Type, Date, Last-Modified, Server, X-Cache), and the body (HTML document). The packet bytes pane on the right shows the raw data of the response, including the status line and the body content.

Packet 7018: HTTP response from 192.229.211.108 to 192.168.86.38. The response includes a Set-Cookie header.

Packet details:

- 200 OK
- Accept-Ranges: bytes
- Age: 6388
- Cache-Control: max-age=7200
- Content-Type: application/ocsp-response
- Date: Fri, 06 Dec 2024 22:06:36 GMT
- Last-Modified: Fri, 06 Dec 2024 20:20:08 GMT
- Server: ECACC (lac/55CC)
- X-Cache: HIT
- Content-Length: 1587

Packet bytes (hex):

```
0000  a6 fb c4 5f 05 0b b0 6a 41 ac 96 c3 08 00 45 00 ...
0010  01 7a a1 b8 00 00 36 06 36 a5 c0 e5 d3 6c c0 a8 ...
0020  56 26 00 50 ff 2c ba 49 f7 c7 66 f4 6e 03 80 18 ...
0030  00 83 5e 40 00 00 01 01 08 0a 6f fa 11 3d 5d ce ...
0040  9e ef 07 02 01 16 1c 68 74 7a 70 73 3a 2f 2f 77 ...
0050  77 77 2a 64 69 67 69 63 65 72 74 2a 63 6f 6d 2f ...
0060  43 50 53 30 0f 06 09 2b 06 01 05 05 07 30 01 05 ...
0070  04 02 05 00 30 0d 06 09 2a 86 48 86 f7 0d 01 01 ...
0080  0b 05 00 03 82 01 01 00 6c 27 4e d0 1f 7d c9 51 ...
0090  d4 66 f9 4c c7 a7 d0 1c e2 f8 f9 82 04 7a d2 aa ...
00a0  38 5e 3c 4a 81 b0 b2 d2 c0 13 7c 64 09 46 8a 30 ...
00b0  1f d8 b6 aa f9 54 56 98 69 18 31 ba 74 ef e8 69 ...
00c0  95 52 1a 0d bc c1 5d c0 6b ac 77 4e 42 3b 53 08 ...
00d0  ec de 80 66 40 a6 36 78 27 a6 7a 4f 06 c6 ac ba ...
00e0  43 53 b2 b3 79 6b 2f 32 5c 55 a8 fd 18 13 d1 cd ...
00f0  19 8a 06 32 a3 a4 5b 76 4e 60 70 63 63 4b 26 54 ...
0100  3f 9d 77 6b 83 39 eb cc 83 56 7a ac 12 8a d1 48 ...
0110  2b 39 38 a5 fb 17 4c f8 b2 77 45 c7 a3 64 bc 4f ...
0120  34 4e ff 68 1e 81 1f 97 86 6e 92 29 b7 b8 9a 9a ...
0130  fd ad c2 40 b2 a9 ed 9c 1b da 52 a3 00 04 a0 30 ...
0140  7c 16 6d aa f1 49 96 0c 68 e8 f1 8b d8 e7 03 a7 ...
0150  a9 7e 30 dc 25 b3 c7 d8 b8 a4 82 3b 8a 63 7f 1d ...
0160  12 79 f4 21 ed 95 6c 17 36 22 14 ef e0 11 b7 e7 ...
0170  68 ca 55 2a e9 a0 9f f3 72 0e 45 bb f8 d8 99 ca ...
0180  db 20 b0 12 2e 68 04 3f
```

In the GET response we can see the set cookies