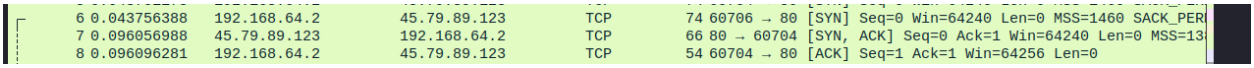
Joshua Grossman

Professor Ondich

CS338: Computer Security

The browser starts by sending a synchronization packet in an attempt to connect with the server. The server acknowledges reception of the packet and sends one in return, thus allowing two-way communication between the browser and server. The browser then sends an acknowledgement packet back to the server to confirm the connection. This is what that looks like:

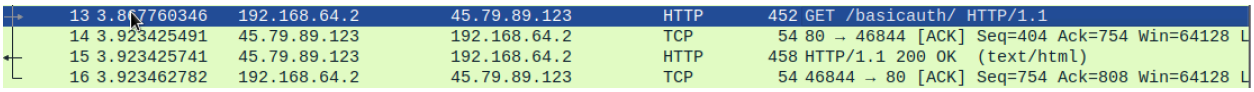


In this screenshot, the 192 (client) IP and server (45) IP are setting up the connection.

Now, the browser asks the server to access the contents of http://jeffondich.com/basicauth/ (GET /basicauth/ HTTP/1.1). The server then acknowledges this request but the server denies the request because because the browser has not provided any authorizing information. The server sends a '401 Unauthorized' message and waits for a response from the browser.

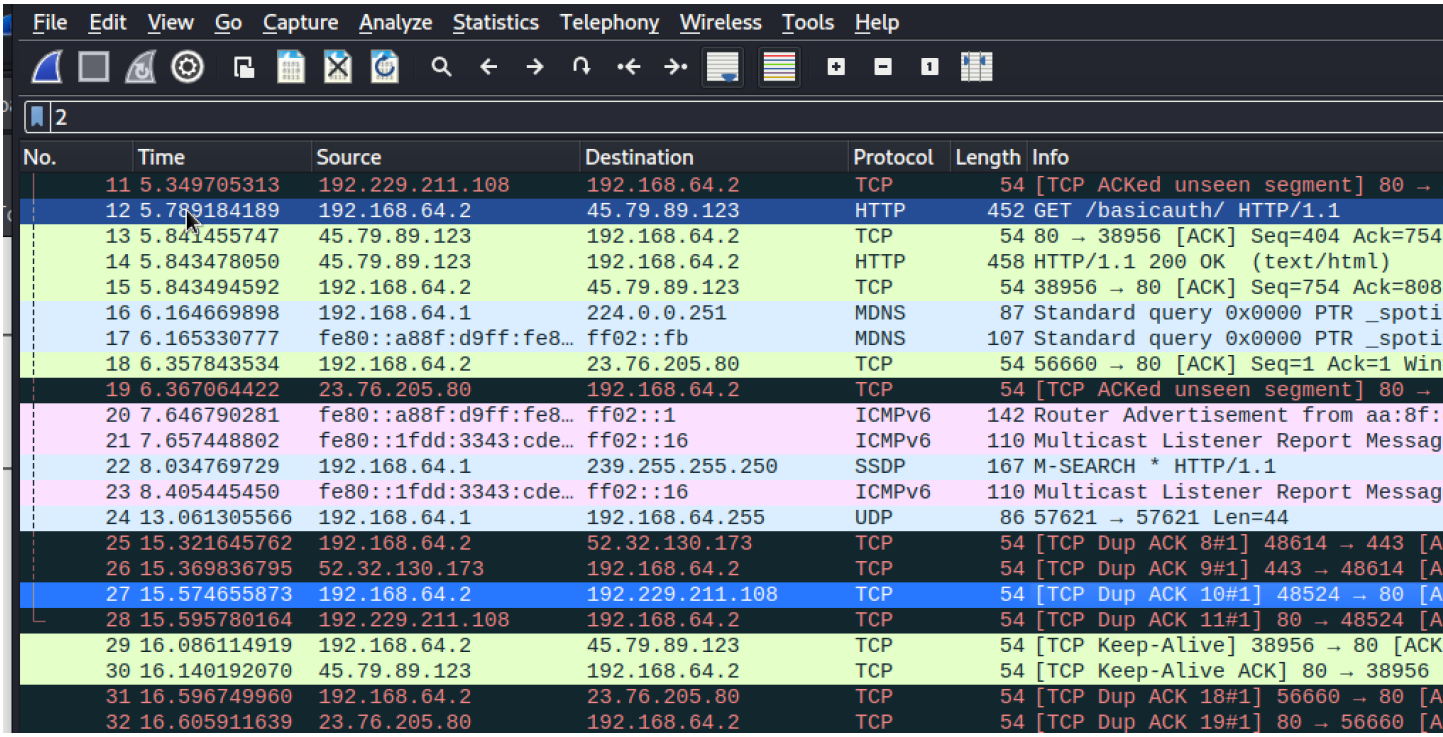
The browser now understands to prompt the user for credentials and sends along another request. We can see in the screenshot below, in the GET request information, that the Authorization section is followed by a hash-like string. This string is the encoded username and password, which have been encoded using UTF-8, or base64.

The encoding is then sent from the browser to the server for authentication. Next, the server acknowledges the GET request. The server then sends back a 200 OK code with the website information. This includes the HTML and the secret files. After receiving all of the information, the browser sends an acknowledgement to the server. This looks like:



Finally, the browser sends one more request, asking the favicon for the website. However, this is unsuccessful. After acknowledging this request, the server searches but does not find anything, and in turn sends back a 404 Not Found which the browser acknowledges. (I had this in an initial run of this but after shutting down my VM once I could not replicate [insert screenshot

here]


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
2
No. Time Source Destination Protocol Length Info
11 5.349705313 192.229.211.108 192.168.64.2 TCP 54 [TCP ACKed unseen segment] 80 →
12 5.789184189 192.168.64.2 45.79.89.123 HTTP 452 GET /basicauth/ HTTP/1.1
13 5.841455747 45.79.89.123 192.168.64.2 TCP 54 80 → 38956 [ACK] Seq=404 Ack=754
14 5.843478050 45.79.89.123 192.168.64.2 HTTP 458 HTTP/1.1 200 OK (text/html)
15 5.843494592 192.168.64.2 45.79.89.123 TCP 54 38956 → 80 [ACK] Seq=754 Ack=808
16 6.164669898 192.168.64.1 224.0.0.251 MDNS 87 Standard query 0x0000 PTR _spoti
17 6.165330777 fe80::a88f:d9ff:fe8… ff02::fb MDNS 107 Standard query 0x0000 PTR _spoti
18 6.357843534 192.168.64.2 23.76.205.80 TCP 54 56660 → 80 [ACK] Seq=1 Ack=1 Win
19 6.367064422 23.76.205.80 192.168.64.2 TCP 54 [TCP ACKed unseen segment] 80 →
20 7.646790281 fe80::a88f:d9ff:fe8… ff02::1 ICMPv6 142 Router Advertisement from aa:8f:
21 7.657448802 fe80::1fdd:3343:cde… ff02::16 ICMPv6 110 Multicast Listener Report Messag
22 8.034769729 192.168.64.1 239.255.255.250 SSDP 167 M-SEARCH * HTTP/1.1
23 8.405445450 fe80::1fdd:3343:cde… ff02::16 ICMPv6 110 Multicast Listener Report Messag
24 13.061305566 192.168.64.1 192.168.64.255 UDP 86 57621 → 57621 Len=44
25 15.321645762 192.168.64.2 52.32.130.173 TCP 54 [TCP Dup ACK 8#1] 48614 → 443 [A
26 15.369836795 52.32.130.173 192.168.64.2 TCP 54 [TCP Dup ACK 9#1] 443 → 48614 [A
27 15.574655873 192.168.64.2 192.229.211.108 TCP 54 [TCP Dup ACK 10#1] 48524 → 80 [A
28 15.595780164 192.229.211.108 192.168.64.2 TCP 54 [TCP Dup ACK 11#1] 80 → 48524 [A
29 16.086114919 192.168.64.2 45.79.89.123 TCP 54 [TCP Keep-Alive] 38956 → 80 [ACK
30 16.140192070 45.79.89.123 192.168.64.2 TCP 54 [TCP Keep-Alive ACK] 80 → 38956
31 16.596749960 192.168.64.2 23.76.205.80 TCP 54 [TCP Dup ACK 18#1] 56660 → 80 [A
32 16.605911639 23.76.205.80 192.168.64.2 TCP 54 [TCP Dup ACK 19#1] 80 → 56660 [A