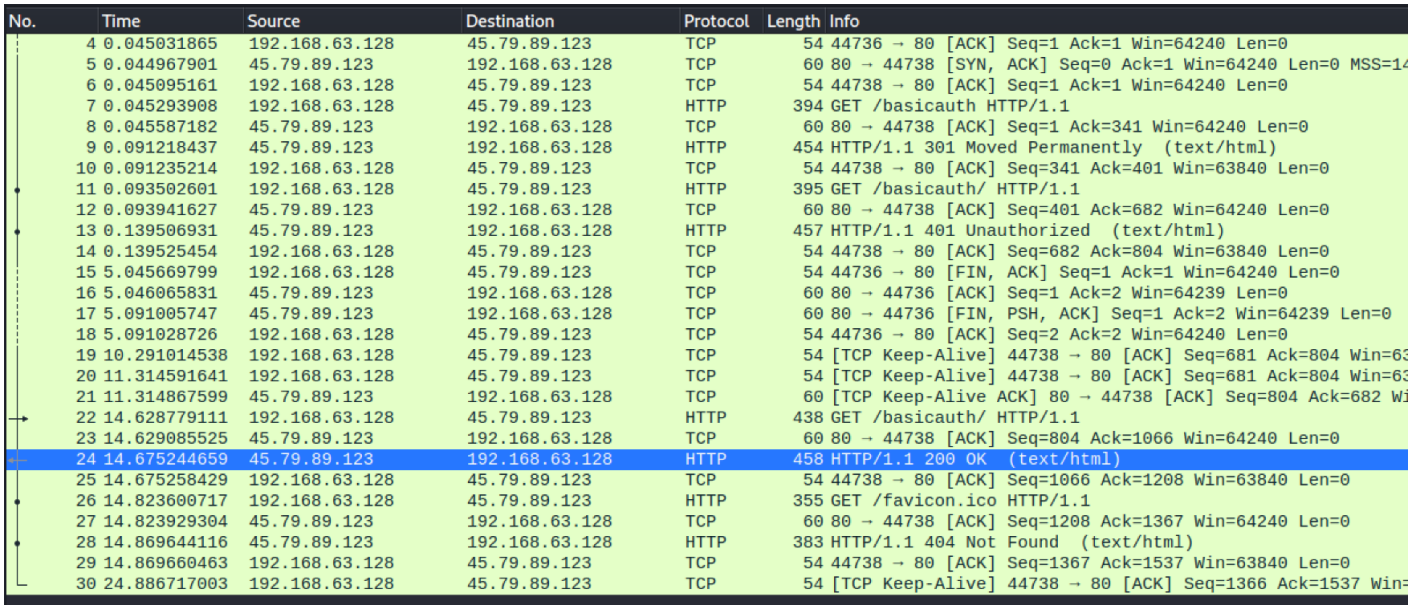
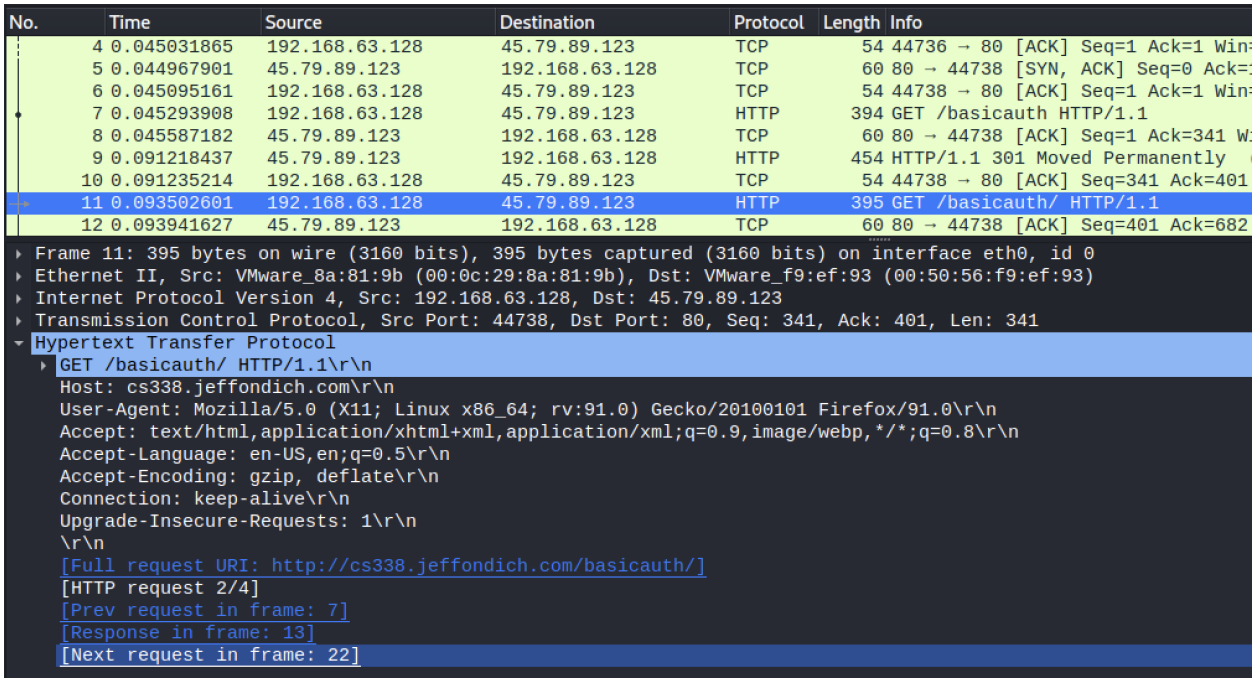
Zack Johnson & Angela Ellis
CS 338: HTTP's Basic Authentication
8 April 2022



No.	Time	Source	Destination	Protocol	Length	Info
4	0.045031865	192.168.63.128	45.79.89.123	TCP	54	44736 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
5	0.044967901	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=14
6	0.045095161	192.168.63.128	45.79.89.123	TCP	54	44738 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
7	0.045293908	192.168.63.128	45.79.89.123	HTTP	394	GET /basicauth HTTP/1.1
8	0.045587182	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [ACK] Seq=1 Ack=341 Win=64240 Len=0
9	0.091218437	45.79.89.123	192.168.63.128	HTTP	454	HTTP/1.1 301 Moved Permanently (text/html)
10	0.091235214	192.168.63.128	45.79.89.123	TCP	54	44738 → 80 [ACK] Seq=341 Ack=401 Win=63840 Len=0
11	0.093502601	192.168.63.128	45.79.89.123	HTTP	395	GET /basicauth/ HTTP/1.1
12	0.093941627	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [ACK] Seq=401 Ack=682 Win=64240 Len=0
13	0.139506931	45.79.89.123	192.168.63.128	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
14	0.139525454	192.168.63.128	45.79.89.123	TCP	54	44738 → 80 [ACK] Seq=682 Ack=804 Win=63840 Len=0
15	5.045669799	192.168.63.128	45.79.89.123	TCP	54	44736 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
16	5.046065831	45.79.89.123	192.168.63.128	TCP	60	80 → 44736 [ACK] Seq=1 Ack=2 Win=64239 Len=0
17	5.091005747	45.79.89.123	192.168.63.128	TCP	60	80 → 44736 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
18	5.091028726	192.168.63.128	45.79.89.123	TCP	54	44736 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
19	10.291014538	192.168.63.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 44738 → 80 [ACK] Seq=681 Ack=804 Win=63
20	11.314591641	192.168.63.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 44738 → 80 [ACK] Seq=681 Ack=804 Win=63
21	11.314867599	45.79.89.123	192.168.63.128	TCP	60	[TCP Keep-Alive ACK] 80 → 44738 [ACK] Seq=804 Ack=682 Wi
22	14.628779111	192.168.63.128	45.79.89.123	HTTP	438	GET /basicauth/ HTTP/1.1
23	14.629085525	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [ACK] Seq=804 Ack=1066 Win=64240 Len=0
24	14.675244659	45.79.89.123	192.168.63.128	HTTP	458	HTTP/1.1 200 OK (text/html)
25	14.675258429	192.168.63.128	45.79.89.123	TCP	54	44738 → 80 [ACK] Seq=1066 Ack=1208 Win=63840 Len=0
26	14.823600717	192.168.63.128	45.79.89.123	HTTP	355	GET /favicon.ico HTTP/1.1
27	14.823929304	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [ACK] Seq=1208 Ack=1367 Win=64240 Len=0
28	14.869644116	45.79.89.123	192.168.63.128	HTTP	383	HTTP/1.1 404 Not Found (text/html)
29	14.869660463	192.168.63.128	45.79.89.123	TCP	54	44738 → 80 [ACK] Seq=1367 Ack=1537 Win=63840 Len=0
30	24.886717003	192.168.63.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 44738 → 80 [ACK] Seq=1366 Ack=1537 Win=

Initially, two ports attempt to connect to the server, (44736, 44738). Both connections are made, and later, the connection to port 44736 is ended. The client asks for the page, “GET /basicauth/” (No. 7). First the server responds that the page being requested has been permanently moved (No. 9). The client asks again (No. 11) and the server responds “401 Unauthorized” (No. 13). Throughout these conversations, acknowledges (ACK) follow each request and response from both the client and the server. The client asks the server to Keep-Alive as Zack and I slowly enter the username and password (No. 19-21). After entering the correct information, the client asks again for the /basicauth page. The server acknowledges this request and sends the requested page “200 OK (text/html)” (No. 24). The client acknowledges that it has received the page (No. 25) and asks for the server to stay connected (No. 30) until we close the browser.

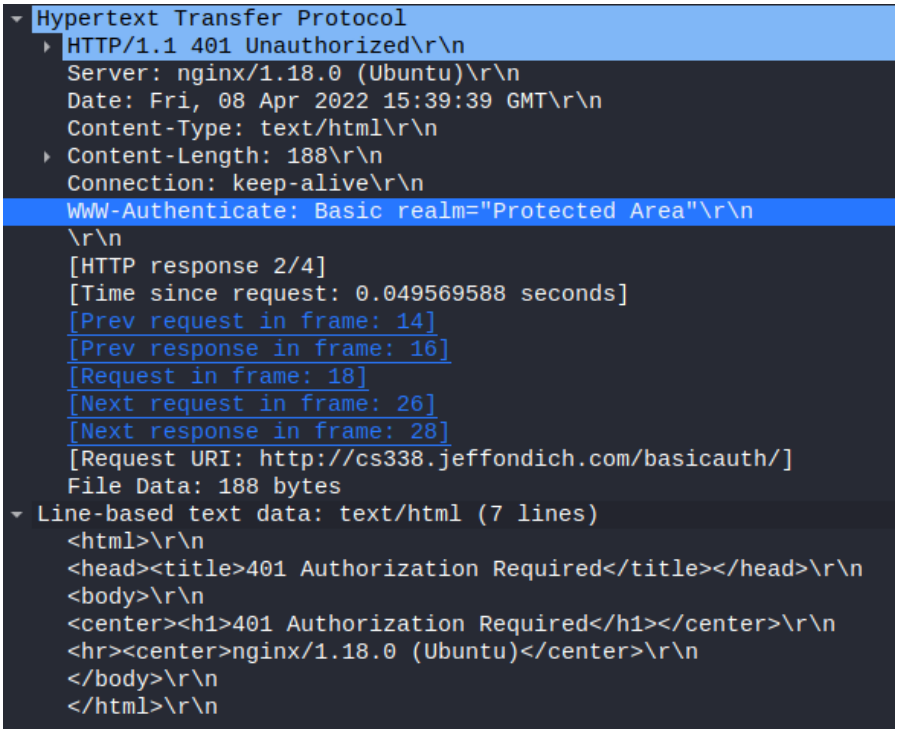
We reference the 'Basic' HTTP Authentication specification throughout, it can be found at, https://datatracker.ietf.org/doc/html/rfc7617.



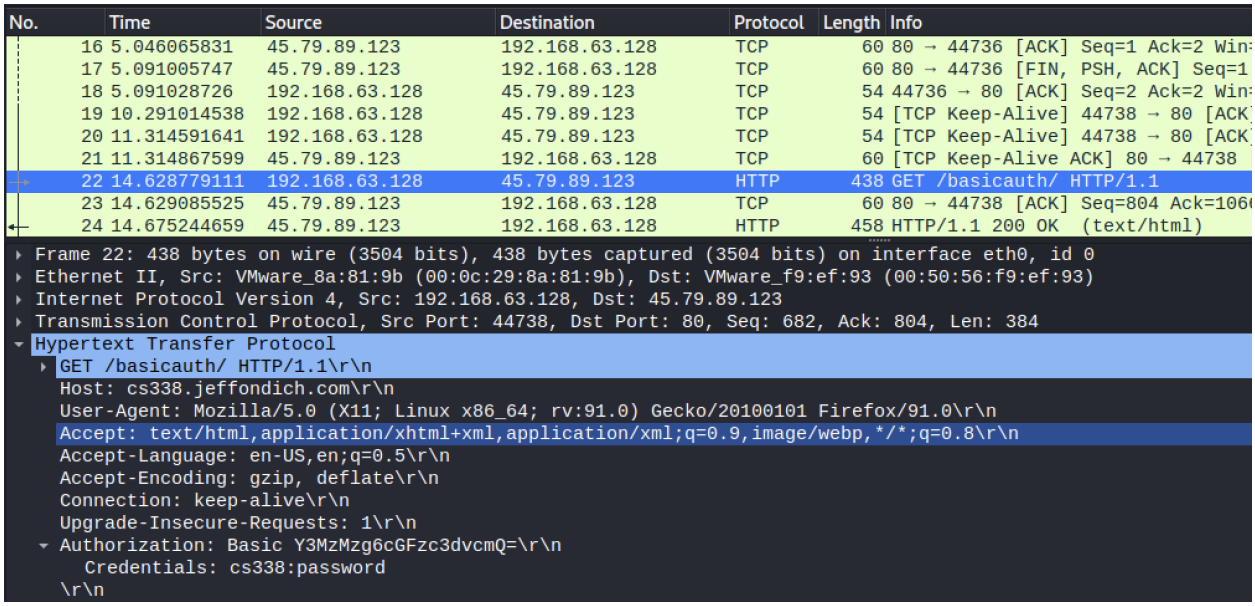
```
No.  Time          Source          Destination     Protocol Length Info
     4 0.045031865 192.168.63.128  45.79.89.123    TCP        54 44736 → 80 [ACK] Seq=1 Ack=1 Win=
     5 0.044967901 45.79.89.123    192.168.63.128  TCP        60 80 → 44738 [SYN, ACK] Seq=0 Ack=1
     6 0.045095161 192.168.63.128  45.79.89.123    TCP        54 44738 → 80 [ACK] Seq=1 Ack=1 Win=
     7 0.045293908 192.168.63.128  45.79.89.123    HTTP      394 GET /basicauth HTTP/1.1
     8 0.045587182 45.79.89.123    192.168.63.128  TCP        60 80 → 44738 [ACK] Seq=1 Ack=341 Wi
     9 0.091218437 45.79.89.123    192.168.63.128  HTTP      454 HTTP/1.1 301 Moved Permanently
    10 0.091235214 192.168.63.128  45.79.89.123    TCP        54 44738 → 80 [ACK] Seq=341 Ack=401
    11 0.093502601 192.168.63.128  45.79.89.123    HTTP      395 GET /basicauth/ HTTP/1.1
    12 0.093941627 45.79.89.123    192.168.63.128  TCP        60 80 → 44738 [ACK] Seq=401 Ack=682
▸ Frame 11: 395 bytes on wire (3160 bits), 395 bytes captured (3160 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_8a:81:9b (00:0c:29:8a:81:9b), Dst: VMware_f9:ef:93 (00:50:56:f9:ef:93)
▸ Internet Protocol Version 4, Src: 192.168.63.128, Dst: 45.79.89.123
▸ Transmission Control Protocol, Src Port: 44738, Dst Port: 80, Seq: 341, Ack: 401, Len: 341
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth/]
    [HTTP request 2/4]
    [Prev request in frame: 7]
    [Response in frame: 13]
    [Next request in frame: 22]
```

Diving deeper into HTTP portions of this story, let's look at Frame 11. In this frame, the client requests the page with the normal headers like Host, User-Agent, etc. Specifically, it is missing the Authorization Header.

The server responds by sending "401 Unauthorized." This frame contains the HTML stating that the page requires authorization. Also, the HTTP header contains the information outlined in the spec (RFC 7617). For example, it has the header "WWW-Authenticate" with the correct parameter "Basic realm" which in this case is called, "Protected Area." The spec outlines that the client needs to construct the username and password (RFC 7617, p. 4). Firefox automatically creates a pop up for us to enter in this information, without displaying the HTML.



```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 401 Unauthorized\r\n
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Fri, 08 Apr 2022 15:39:39 GMT\r\n
    Content-Type: text/html\r\n
  ▸ Content-Length: 188\r\n
    Connection: keep-alive\r\n
    WWW-Authenticate: Basic realm="Protected Area"\r\n
    \r\n
    [HTTP response 2/4]
    [Time since request: 0.049569588 seconds]
    [Prev request in frame: 14]
    [Prev response in frame: 16]
    [Request in frame: 18]
    [Next request in frame: 26]
    [Next response in frame: 28]
    [Request URI: http://cs338.jeffondich.com/basicauth/]
    File Data: 188 bytes
▾ Line-based text data: text/html (7 lines)
    <html>\r\n
    <head><title>401 Authorization Required</title></head>\r\n
    <body>\r\n
    <center><h1>401 Authorization Required</h1></center>\r\n
    <hr><center>nginx/1.18.0 (Ubuntu)</center>\r\n
    </body>\r\n
    </html>\r\n
```



No.	Time	Source	Destination	Protocol	Length	Info
16	5.046065831	45.79.89.123	192.168.63.128	TCP	60	80 → 44736 [ACK] Seq=1 Ack=2 Win=
17	5.091005747	45.79.89.123	192.168.63.128	TCP	60	80 → 44736 [FIN, PSH, ACK] Seq=1
18	5.091028726	192.168.63.128	45.79.89.123	TCP	54	44736 → 80 [ACK] Seq=2 Ack=2 Win=
19	10.291014538	192.168.63.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 44738 → 80 [ACK]
20	11.314591641	192.168.63.128	45.79.89.123	TCP	54	[TCP Keep-Alive] 44738 → 80 [ACK]
21	11.314867599	45.79.89.123	192.168.63.128	TCP	60	[TCP Keep-Alive ACK] 80 → 44738
22	14.628779111	192.168.63.128	45.79.89.123	HTTP	438	GET /basicauth/ HTTP/1.1
23	14.629085525	45.79.89.123	192.168.63.128	TCP	60	80 → 44738 [ACK] Seq=804 Ack=106
24	14.675244659	45.79.89.123	192.168.63.128	HTTP	458	HTTP/1.1 200 OK (text/html)

```
▸ Frame 22: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_8a:81:9b (00:0c:29:8a:81:9b), Dst: VMware_f9:ef:93 (00:50:56:f9:ef:93)
▸ Internet Protocol Version 4, Src: 192.168.63.128, Dst: 45.79.89.123
▸ Transmission Control Protocol, Src Port: 44738, Dst Port: 80, Seq: 682, Ack: 804, Len: 384
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▾ Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
      Credentials: cs338:password
    \r\n
```

Once we enter the correct username and password, the client formats the information *username:password* and encodes it using Base64, as outlined in the spec (RFC 7617, p. 4) We see the result of this in the header "Authorization" of Frame 22. The HTTP header also contains the byte sequence of the credentials as "Y3MzMzg6cGFzc3dvcmQ=", and Wireshark converts this back to us as "cs338:password." This Base64 string is sent to the server with the request packet for the server to authenticate. As the spec states, this is insecure authentication because the credentials are encoded but not encrypted. This means anyone with packet sniffers anywhere on any machine the packets traverse can have access to this username and password. The implications of this insecurity are covered in section four of the RFC 7617.

Finally, the server responds with "200 OK" and sends the requested page (No. 24). We can see the HTML in the payload of the frame. This packet concludes the Basic Authentication Story.

```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Fri, 08 Apr 2022 15:39:48 GMT\r\n
    Content-Type: text/html\r\n
    Transfer-Encoding: chunked\r\n
    Connection: keep-alive\r\n
    Content-Encoding: gzip\r\n
    \r\n
    [HTTP response 3/4]
    [Time since request: 0.050768226 seconds]
    [Prev request in frame: 18]
    [Prev response in frame: 20]
    [Request in frame: 26]
    [Next request in frame: 30]
    [Next response in frame: 32]
    [Request URI: http://cs338.jeffondich.com/basicauth/]
  ▸ HTTP chunked response
    Content-encoded entity body (gzip): 205 bytes -> 509 bytes
    File Data: 509 bytes
▾ Line-based text data: text/html (9 lines)
    <html>\r\n
    <head><title>Index of /basicauth/</title></head>\r\n
    <body>\r\n
    <h1>Index of /basicauth/</h1><hr><pre><a href="../">../</a>\r\n
    <a href="amateurs.txt">amateurs.txt</a>
    <a href="armed-guards.txt">armed-guards.txt</a>
    <a href="dancing.txt">dancing.txt</a>
    </pre><hr></body>\r\n
    </html>\r\n
```