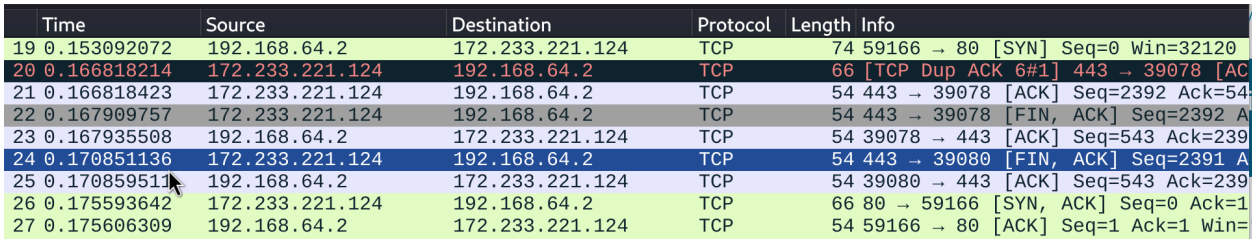
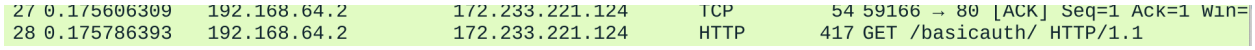
HTTP basic authentication HW

First, the client initiates a TCP handshake by sending to the http://cs338.jeffondich.com server a [SYN] packet (line 19) containing a sequence number. The server responds with a [SYN,ACK] packet (line 26) which acknowledges the client's [SYN] packet, and sends another sequence number. The client also acknowledges the server's sequence number (line 27) by sending an [ACK] packet.



	Time	Source	Destination	Protocol	Length	Info
19	0.153092072	192.168.64.2	172.233.221.124	TCP	74	59166 → 80 [SYN] Seq=0 Win=32120
20	0.166818214	172.233.221.124	192.168.64.2	TCP	66	[TCP Dup ACK 6#1] 443 → 39078 [AC
21	0.166818423	172.233.221.124	192.168.64.2	TCP	54	443 → 39078 [ACK] Seq=2392 Ack=54
22	0.167909757	172.233.221.124	192.168.64.2	TCP	54	443 → 39078 [FIN, ACK] Seq=2392 A
23	0.167935508	192.168.64.2	172.233.221.124	TCP	54	39078 → 443 [ACK] Seq=543 Ack=239
24	0.170851136	172.233.221.124	192.168.64.2	TCP	54	443 → 39080 [FIN, ACK] Seq=2391 A
25	0.170859511	192.168.64.2	172.233.221.124	TCP	54	39080 → 443 [ACK] Seq=543 Ack=239
26	0.175593642	172.233.221.124	192.168.64.2	TCP	66	80 → 59166 [SYN, ACK] Seq=0 Ack=1
27	0.175606309	192.168.64.2	172.233.221.124	TCP	54	59166 → 80 [ACK] Seq=1 Ack=1 Win=

Once the TCP connection is established, the client sends an HTTP GET request to the server (line 28), requesting a file called /basicauth/

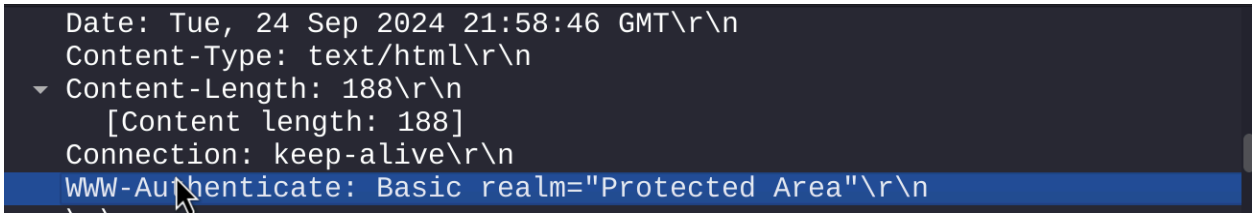


27	0.175606309	192.168.64.2	172.233.221.124	TCP	54	59166 → 80 [ACK] Seq=1 Ack=1 Win=
28	0.175786393	192.168.64.2	172.233.221.124	HTTP	417	GET /basicauth/ HTTP/1.1

The web server (nginx) recognizes that the site is password protected, and accordingly sends the following packet to let the user know that the user is unauthorized, and that authentication is needed.

30	0.205150889	172.233.221.124	192.168.64.2	HTTP	457	HTTP/1.1 401 Unauthorized (text/

Inside this packet, the server also indicates the authentication scheme it expects the user to abide by:



```
Date: Tue, 24 Sep 2024 21:58:46 GMT\r\n
Content-Type: text/html\r\n
Content-Length: 188\r\n
  [Content length: 188]
Connection: keep-alive\r\n
WWW-Authenticate: Basic realm="Protected Area"\r\n
```

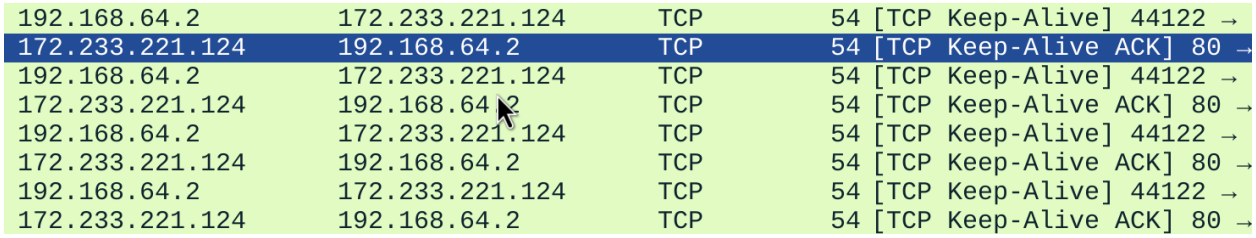
At the WWW-Authenticate response header, the "Basic" indicates that the authentication scheme name is "Basic". This authentication scheme requires the user to provide authentication by sending the correct username and password to the server.[1] A "realm" indicates a collection of resources on the server in which particular users and groups can access the file contents.[2]

[1]https://datatracker.ietf.org/doc/html/rfc7617#section-1
[2]https://docs.oracle.com/cd/E19226-01/820-7627/bnbxm/index.html

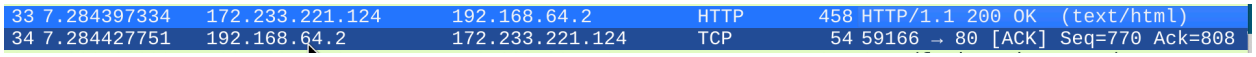
This prompts the user on the client side to provide a username and password. The client browser does not check for password correctness, it has to send it to the server in an appropriate format. Accordingly, the client concatenates the **username** and **password** into the following string: "**username**:**password**" and encode the string with base64.[3] The base64 encoding is put into the "Authorization" request header:

'Basic' indicates that the authentication scheme name is "Basic," and the adjacent messy string of letters is the base64 encoding of the password.[4] Base64 encoding is not an encryption method—anyone who can access the packet contents can easily decode back to the original string using the base64 decoding scheme, which is publicly available.[5] Rather, the username and password are sent in base64 because HTTP header-field values, for some reason, require a character set of printable ASCII characters, and base64 converts anything else into that character set. The important takeaway here is that the Basic authentication scheme is not secure, and base64 changes non-ASCII non-printable characters into something compatible with HTTP format.[6]

After a while, if the client does not send the username and password, both the client and the server send a [keep-alive] packets to each other to prevent the connection from being terminated.



After the correct username and password are sent, the server decodes from base64 to the original **username**:**password** string and checks for correctness. It responds with a 200 OK, along with the contents of the password-protected page.



[3]https://datatracker.ietf.org/doc/html/rfc7617#section-2
[4]Ibid.
[5]https://datatracker.ietf.org/doc/html/rfc7617#section-4
[6]https://stackoverflow.com/questions/13661384/why-base64-in-basic-authentication

The client browser also requests for a file called favicon.ico (line 34), which is the small icon positioned at the left side of the tab. However, because the page does not use such an icon, the server responses with a 404 Not Found (line 35).