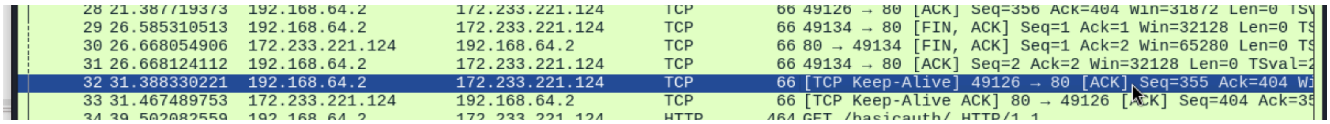
Lauren Caldwell

--Correct password--

- First there's the TCP handshake
- Then there's an HTTP packet that the user asks to get basic authorization
25 21.302155228 192.168.64.2 172.233.221.124 HTTP 421 GET /basicauth/ HTTP/1.1

- Then server then responds and says that the user is unauthorized
27 21.387692708 172.233.221.124 192.168.64.2 HTTP 469 HTTP/1.1 401 Unauthorized (text/html)

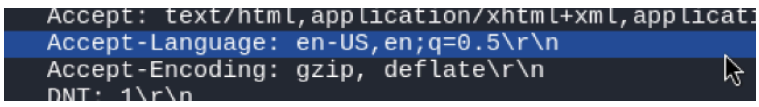
- Next the server attempt to sever the connection and finish the interaction, but the user forces the server to keep the connection alive



- Next the user attempts to get basic authorization again (here the user puts username and password)
34 39.502082559 192.168.64.2 172.233.221.124 HTTP 421 GET /basicauth/ HTTP/1.1

I also think that the browser does the password checking itself and uses an encryption key
When the user puts in the right username and password the authorization header in the GET /basicauth/ packet is as follows:
Authorization: Basic Y3MzMzg6cGFzc3dvcmq= (the random assortment of characters is the encrypted password)

The encryption also comes from the browser



It then uses this encoding language to encode the given password therefore making it unreadable to anyone looking at it through means like WireShark are unable to tell what the password actually is

- Server then accepts username and password and allows access to the server

35 39.588192513 172.233.221.124 192.168.64.2 HTTP 470 HTTP/1.1 200 OK (text/html)

- then there's HTTP packets to get the parts of the servers like normal
- user ends interaction like normal

Observations:
Even when the user is unauthorized to view the server there is still a lot of detail given to the user about the contents of the server. You can get a general sense of how long the document is, what is on it, and how it is organized even without authorization.

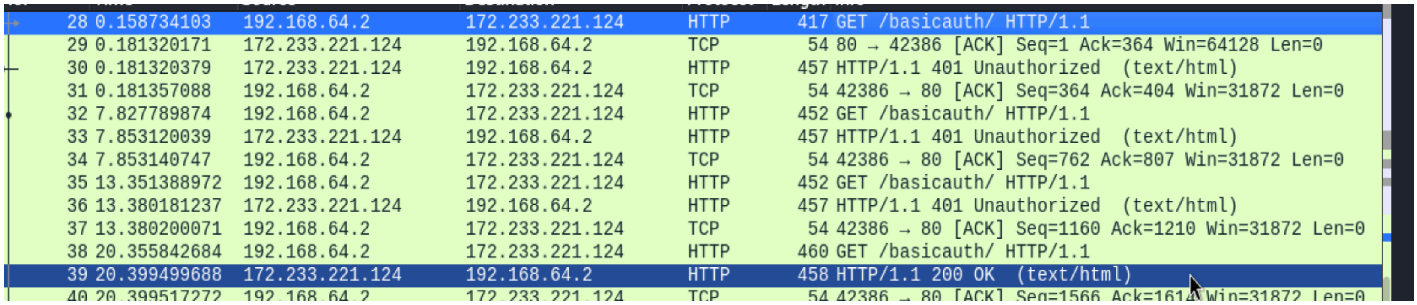
```
HTTP/1.1 401 Unauthorized
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 24 Sep 2024 15:38:48 GMT
Content-Type: text/html
Content-Length: 590
Connection: keep-alive
WWW-Authenticate: Basic realm="Protected Area"

<html>
  <head>
    <title>
      401 Authorization Required
    </title>
  </head>
  <body>
    <center>
      <h1>
        401 Authorization Required
      </h1>
    </center>
    <hr>
    <center>
      nginx/1.18.0 (Ubuntu)
    </center>
  </body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->

```

Even without authorization here you can see that it's an html, its 590 bytes, and the relative structure that the html is with heads and titles etc.

--Incorrect password--

When given an incorrect username or password the process is almost identical to when it was given the correct password except when given the incorrect password instead of getting the 200 OK it instead goes back to the unauthorized packet until you type in the

correct password and username.



28	0.158734103	192.168.64.2	172.233.221.124	HTTP	417	GET /basicauth/ HTTP/1.1
29	0.181320171	172.233.221.124	192.168.64.2	TCP	54	80 → 42386 [ACK] Seq=1 Ack=364 Win=64128 Len=0
30	0.181320379	172.233.221.124	192.168.64.2	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
31	0.181357088	192.168.64.2	172.233.221.124	TCP	54	42386 → 80 [ACK] Seq=364 Ack=404 Win=31872 Len=0
32	7.827789874	192.168.64.2	172.233.221.124	HTTP	452	GET /basicauth/ HTTP/1.1
33	7.853120039	172.233.221.124	192.168.64.2	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
34	7.853140747	192.168.64.2	172.233.221.124	TCP	54	42386 → 80 [ACK] Seq=762 Ack=807 Win=31872 Len=0
35	13.351388972	192.168.64.2	172.233.221.124	HTTP	452	GET /basicauth/ HTTP/1.1
36	13.380181237	172.233.221.124	192.168.64.2	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
37	13.380200071	192.168.64.2	172.233.221.124	TCP	54	42386 → 80 [ACK] Seq=1160 Ack=1210 Win=31872 Len=0
38	20.355842684	192.168.64.2	172.233.221.124	HTTP	460	GET /basicauth/ HTTP/1.1
39	20.399499688	172.233.221.124	192.168.64.2	HTTP	458	HTTP/1.1 200 OK (text/html)
40	20.399517272	192.168.64.2	172.233.221.124	TCP	54	42386 → 80 [ACK] Seq=1566 Ack=1614 Win=31872 Len=0

You can also see the encrypted incorrect passwords under the authorization header

```
2 Host: cs338.jeffondich.com
3 Cache-Control: max-age=0
4 Authorization: Basic d3V0OndldA==
5 Accept-Language: en-US
```