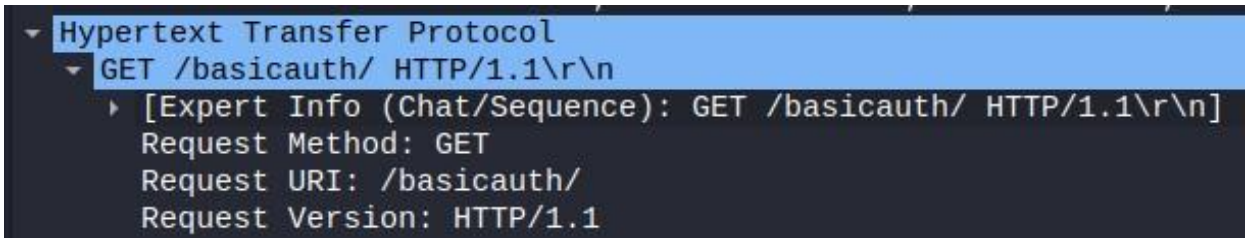
Client-Server Communication

1. What queries are sent from the browser, and what responses does it receive?
    1. TCP Handshake:
        a. The client sends a "SYN" packet to the server to initiate the connection request.
        b. The server responds with a "SYN-ACK" packet, acknowledging the request and signaling its readiness to establish the connection.
        c. The client then sends an "ACK" packet to confirm the connection, and data exchange can begin.

Source	Destination	Protocol	Length	Info
192.168.112.128	45.79.89.123	TCP	74	51554 → 80 [SYN] Seq=0
45.79.89.123	192.168.112.128	TCP	60	80 → 51554 [SYN, ACK]
192.168.112.128	45.79.89.123	TCP	54	51554 → 80 [ACK] Seq=1

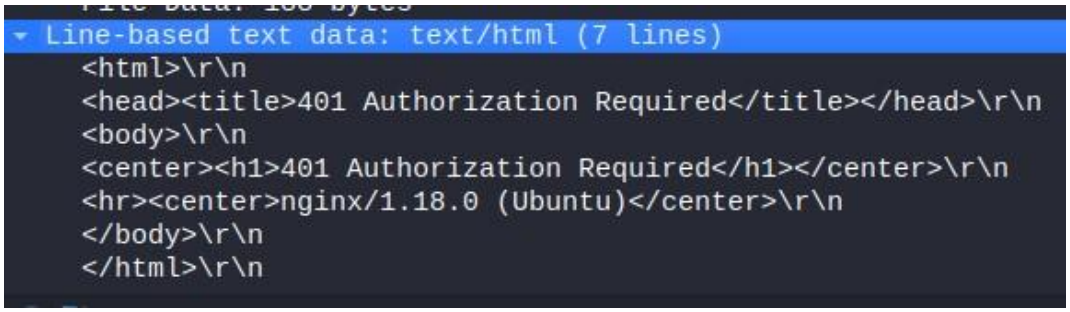
2. The client sends an HTTP Get request to the server:
    a. 4 192.168.112.128 45.79.89.123 HTTP GET /basicauth/ HTTP/1.1



3. The server acknowledges the request and sends an HTTP response to the client.

192.168.112.128	45.79.89.123	HTTP	408	GET /basicauth/ HTTP/1.1
45.79.89.123	192.168.112.128	TCP	60	80 → 51554 [ACK] Seq=1 Ack=355 Win=642
45.79.89.123	192.168.112.128	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
192.168.112.128	45.79.89.123	TCP	54	51554 → 80 [ACK] Seq=355 Ack=404 Win=6

4. Here is the content of the server's response which basically asks for authorization

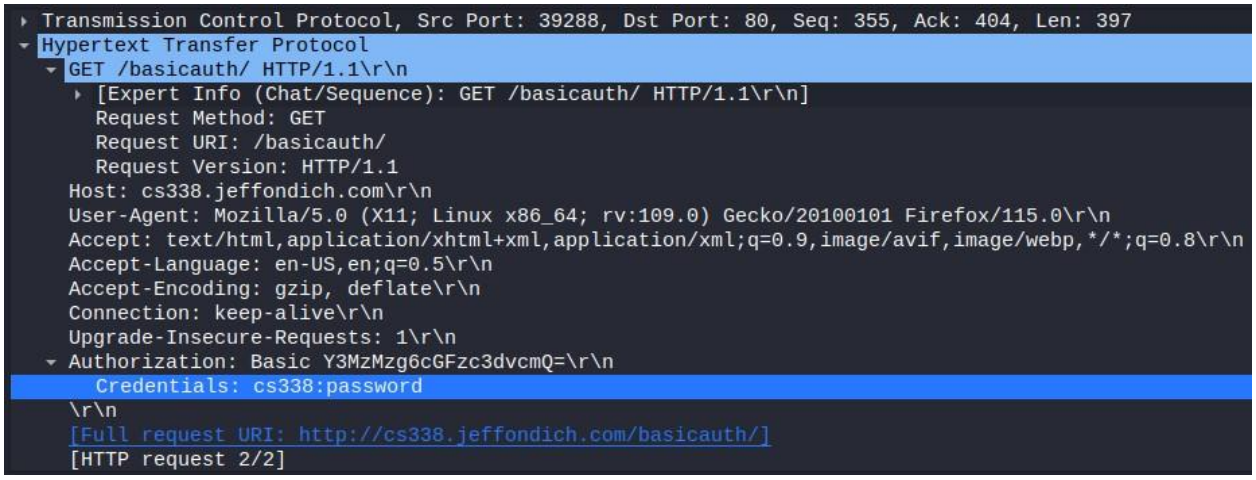


5. The client acknowledges the response of the server.

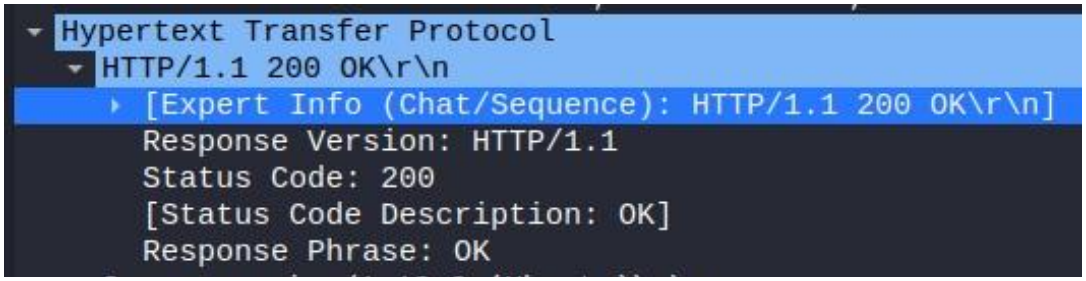
6. The client enters the authorization and sends another HTTP Get request to the server.

```
192.168.112.128    45.79.89.123       HTTP     451 GET /basicauth/ HTTP/1.1
45.79.89.123       192.168.112.128    TCP       60 80 → 39288 [ACK] Seq=404 Ack=752
45.79.89.123       192.168.112.128    HTTP     458 HTTP/1.1 200 OK  (text/html)
192.168.112.128    45.79.89.123       TCP       54 39288 → 80 [ACK] Seq=752 Ack=808
```

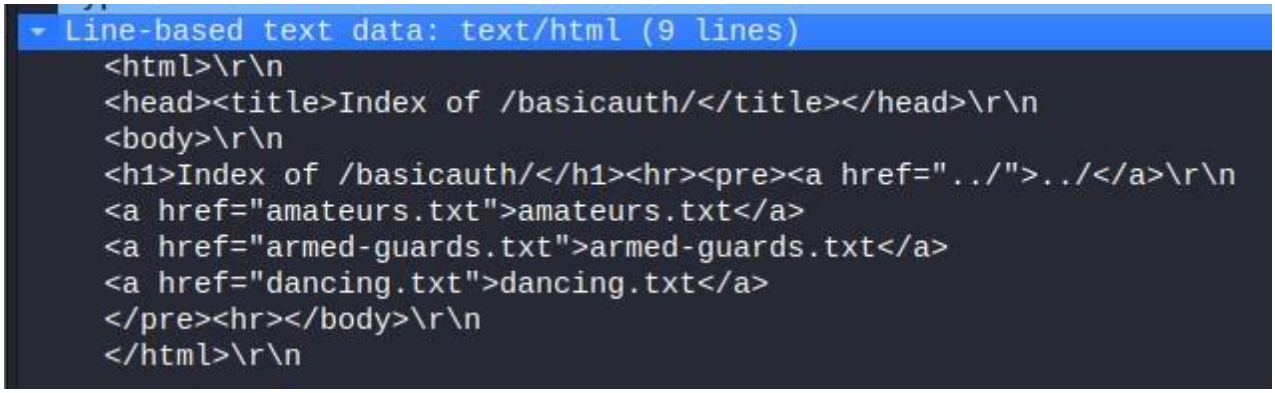
7. Here is the GET request which also includes the credentials: username and password.



8. The TCP protocol verifies the credentials and sends an acknowledgment.
9. The server responds with code 200 to the client to signal the approval of get request.



10. Here is the full content of the HTML which includes the files in the secret folder.



11. The client acknowledges the response from the server through the TCP protocol.

2. Is the password sent by the browser to the server, or does the browser somehow do the password checking itself?

a. I think the password is sent by the browser to the server because the server sends an acknowledgment after the client gets the credentials from the user and sends it to the server.

3. If the former, is the password sent in clear text or is it encrypted or something else?

a. Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
b. Credentials: cs338:password
c. There is a key, which indicates that the password is encrypted.

4. If it's encrypted, where did the encryption key come from?

a. Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
b. The key seems to be generated through base64

5. How does what you observe via Wireshark connect to the relevant sections of the HTTP and HTTP Basic Authentication specification documents? Etc.

a. To receive authorization, the client
    i. obtains the user-id and password from the user,
    ii. constructs the user-pass by concatenating the user-id, a single colon (":") character, and the password,
    iii. encodes the user pass into an octet sequence
    iv. and obtains the basic credentials by encoding this octet sequence using Base64 into a sequence of US-ASCII characters. Below is the example from Wireshark that matches the outcome of the process described above:
        1. Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
        2. Credentials: cs338:password
b. HTTP Method definitions:
    i. “The GET method requests the transfer of a current selected representation for the target resource. GET is the primary mechanism of information retrieval and the focus of almost all performance optimizations. Hence, when people speak of retrieving some identifiable information via HTTP, they are generally referring to making a GET request.”

6. Citations:

a. https://datatracker.ietf.org/doc/html/rfc7617#section-2
b. https://datatracker.ietf.org/doc/html/rfc7231#section-4.3