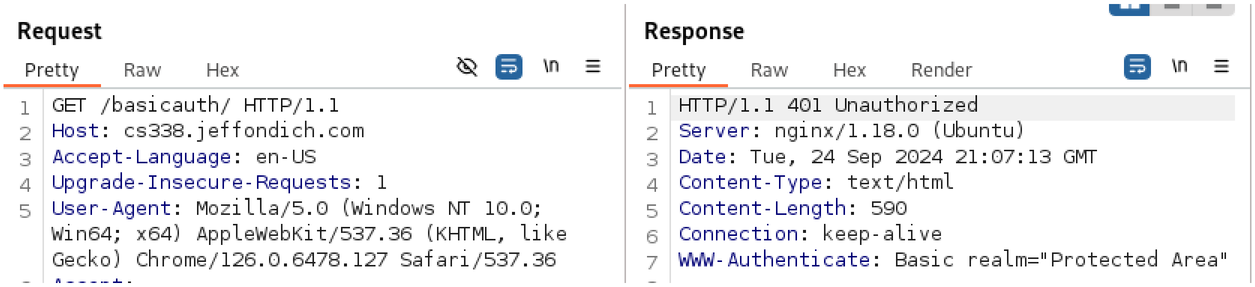
Liam K

# HTTP's Basic Authentication Assignment

## Order of events:

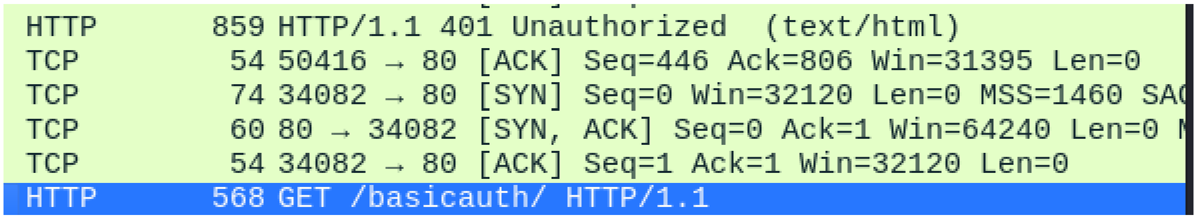
1. The three way TCP handshake is initiated between client and server, and the connection is good to go.
2. My computer, the client, sends Jeff's server a GET request asking for the /basicauth/ page
3. After acknowledging receiving the request, the server responds with a **401 error**, signifying we have been denied access because we require authentication. The response also specifies what type of authentication the server requires with the **WWW-Authenticate header**. In this case it is asking for a Basic auth token.



```
GET /basicauth/ HTTP/1.1
Host: cs338.jeffondich.com
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/126.0.6478.127 Safari/537.36
```

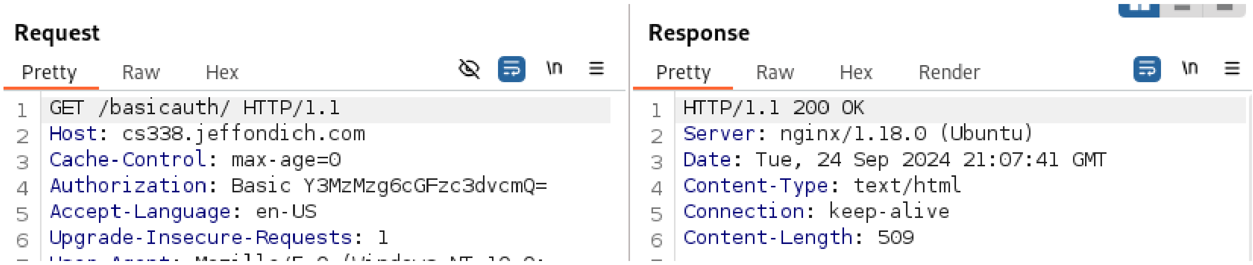
```
HTTP/1.1 401 Unauthorized
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 24 Sep 2024 21:07:13 GMT
Content-Type: text/html
Content-Length: 590
Connection: keep-alive
WWW-Authenticate: Basic realm="Protected Area"
```

4. After forwarding this request, the browser prompts me for credentials.
5. After entering the credentials, my computer initiates another TCP handshake (…I think) on a new port (34082 instead of 50416).bu



```
HTTP      859 HTTP/1.1 401 Unauthorized  (text/html)
TCP        54 50416 → 80 [ACK] Seq=446 Ack=806 Win=31395 Len=0
TCP        74 34082 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SA
TCP        60 80 → 34082 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
TCP        54 34082 → 80 [ACK] Seq=1 Ack=1 Win=32120 Len=0
HTTP      568 GET /basicauth/ HTTP/1.1
```

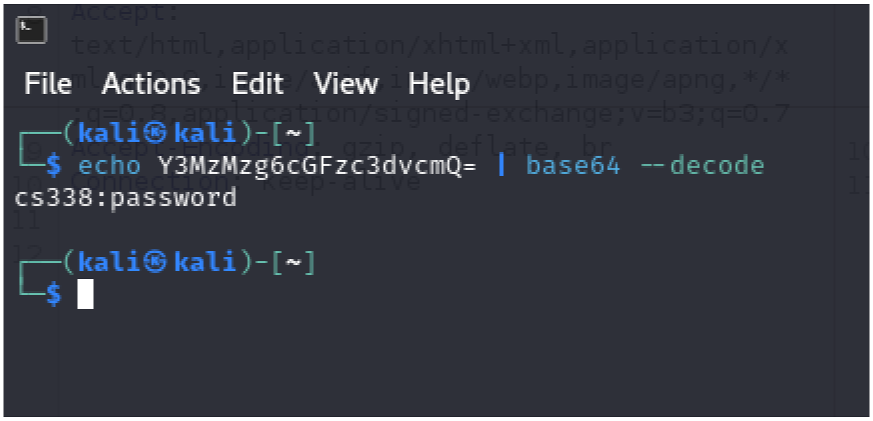
6. My computer then submits the same previous GET request, but with the added **authorization header**, which contains the username and password concatenated together with a colon. Therefore, the browser does not handle the verification of the password because it is passed off to the server to validate before sending back the HTML.



```
GET /basicauth/ HTTP/1.1
Host: cs338.jeffondich.com
Cache-Control: max-age=0
Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 24 Sep 2024 21:07:41 GMT
Content-Type: text/html
Connection: keep-alive
Content-Length: 509
```

The equals character at the end of authorization string signifies that the password is encoded in base64, which is easily reversed back into normal text.



```
┌──(kali㉿kali)-[~]
└─$ echo Y3MzMzg6cGFzc3dvcmQ= | base64 --decode
cs338:password

┌──(kali㉿kali)-[~]
└─$ 
```

Since we are not encrypting anything, this information would be available to any nefarious individuals (or multiples) monitoring the network. This should not be a problem if you're using https since the entire message (including the base64 representation) will be hashed, which is significantly more difficult for an attacker to crack.

Consulted:
https://datatracker.ietf.org/doc/html/rfc7617
https://en.wikipedia.org/wiki/Basic_access_authentication