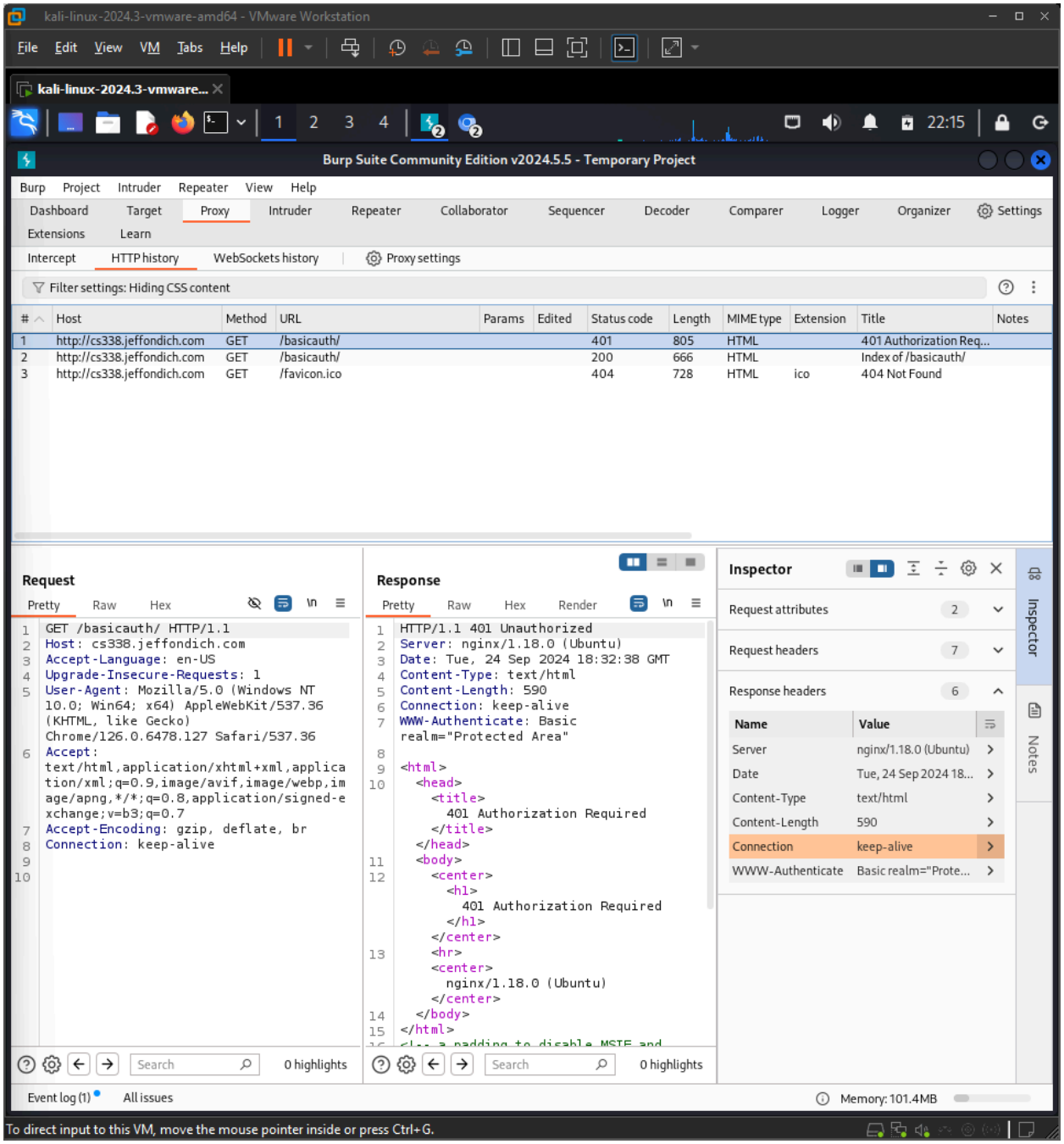
Kaung Thant (John) Win
CS338 - Computer Security

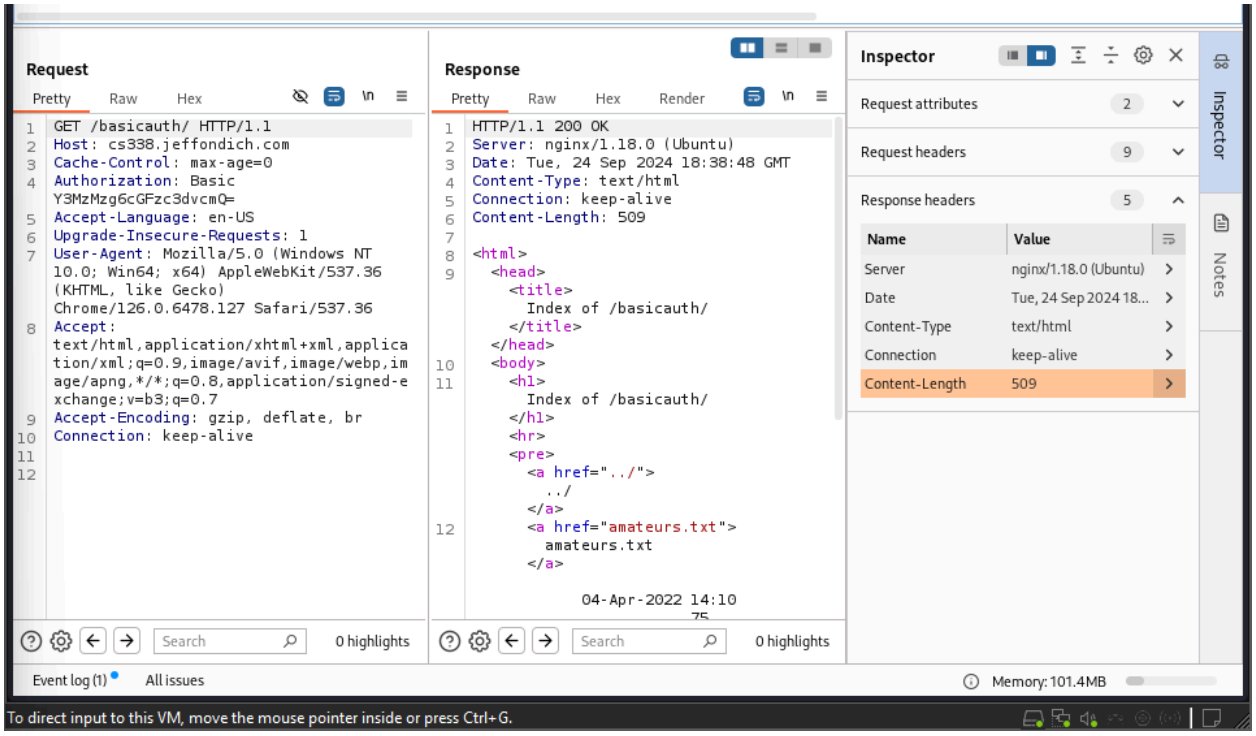
1. As per all HTTP requests, when http://cs338.jeffondich.com/basicauth/ is typed into the URL, the browser sends a “GET /basicauth/ HTTP/1.1” request to the server.
2. However, because the user (me) has not been logged in yet, the server returns a “HTTP/1.1 401 Unauthorized” response to the browser.
    a. The body of the response includes “401 Authorization Required”.
    b. It also includes the header “WWW-Authenticate” with value “Basic realm=”Protected Area””.
        i. WWW-Authenticate provides the authentication schemes and parameters of the requested resource (HTTP Working Group).
        ii. A server sending a 401 Unauthorized response is required to send a WWW-Authenticate header with at least one challenge (HTTP Working Group).
        iii. “Basic realm=”Protected Area”” is a challenge. “Basic” refers to the basic authentication schema which transmits credentials as user ID/password pairs encoded using base64 (mdn web docs). “realm” specifies what the scope of protection is. In this case, it is simply a protected area (mdn web docs).



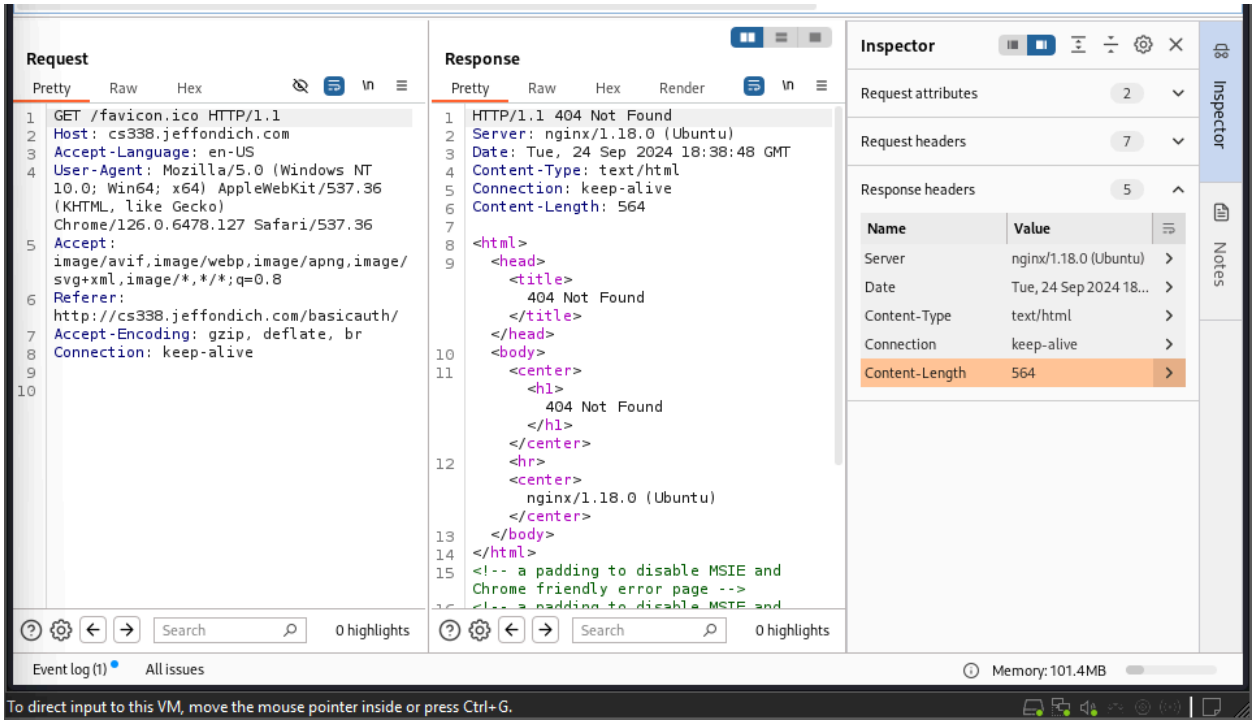
3. Once I log in using the provided credentials, the browser sends another GET request to "/basicauth/".
    a. However, this time the request body includes two additional headers: Cache-Control and Authorization
        i. Cache-Control controls how the response of a request is cached. There are many different types of caches, including private caches and shared caches (mdn web docs). In the example, the value is "max-age = 0" which means the maximum age of the cache is 0 seconds, which means the cache is not stored at all.

ii. According to the HTTP Working Group, the “Authorization” header allows a client to authenticate itself to the server after it has received a 401 response, “which usually comes after a request without credentials” (mdn web docs). The value of “Authorization” contains authentication information about the client specific to the realm being requested. In the current example, the value is “Basic Y3MzMzg6cGFzc3dvcmQ=” where the realm is “basic” and the encoded information is “Y3MzMzg6cGFzc3dvcmQ=”. The information is base64-encoded text, where the format is <userID>:<password> (mdn web docs). If we decode our current text, we get “cs338:password”!

1. According to mdn web docs, the authentication itself happens on the server and the response from the server only contains information on whether the authentication was a success or not.



4. The browser also sends the automatic request of “GET /favicon.ico HTTP/1.1”, which however doesn’t exist on the server and is responded with “HTTP/1.1 404 Not Found”.



Additional Notes using Wireshark

1. As we have previously seen using BurpSuite, the client sends a TCP request to the server to set up a connection, after which the connection is maintained and a “401 Unauthorized” response is received from the server.

```
 1 0.000000000   192.168.126.128   172.233.221.124   TCP     74 56360 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSv…
 2 0.023310210   172.233.221.124   192.168.126.128   TCP     60 80 → 56360 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 3 0.023727500   192.168.126.128   172.233.221.124   TCP     54 56360 → 80 [ACK] Seq=1 Ack=1 Win=32120 Len=0
 4 0.025480429   192.168.126.128   172.233.221.124   HTTP    4… GET /basicauth/ HTTP/1.1
 5 0.025785499   172.233.221.124   192.168.126.128   TCP     60 80 → 56360 [ACK] Seq=1 Ack=446 Win=64240 Len=0
 6 0.052180940   172.233.221.124   192.168.126.128   HTTP    8… HTTP/1.1 401 Unauthorized  (text/html)
 7 0.052218758   192.168.126.128   172.233.221.124   TCP     54 56360 → 80 [ACK] Seq=446 Ack=806 Win=31395 Len=0
```

2. I tested how the server would respond when wrong credentials were provided. A “401 Unauthorized” response is once again received. However, this request is made through a different TCP connection. This is identified by the different source port “41234”. This connection is kept alive.

```
 8 14.667416335  192.168.126.128   172.233.221.124   TCP     74 41234 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSv…
 9 14.688935994  172.233.221.124   192.168.126.128   TCP     60 80 → 41234 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
10 14.688990816  192.168.126.128   172.233.221.124   TCP     54 41234 → 80 [ACK] Seq=1 Ack=1 Win=32120 Len=0
11 14.689352792  192.168.126.128   172.233.221.124   HTTP    5… GET /basicauth/ HTTP/1.1
12 14.689531904  172.233.221.124   192.168.126.128   TCP     60 80 → 41234 [ACK] Seq=1 Ack=507 Win=64240 Len=0
13 14.712753610  172.233.221.124   192.168.126.128   HTTP    8… HTTP/1.1 401 Unauthorized  (text/html)
14 14.712781362  192.168.126.128   172.233.221.124   TCP     54 41234 → 80 [ACK] Seq=507 Ack=806 Win=31395 Len=0
```

3. When the correct credentials are provided, the same “authentication” source port is used again, to which a “200 OK” response is sent and the user authorized to access the documents.

```
15 20.257848844  192.168.126.128   172.233.221.124   HTTP    5… GET /basicauth/ HTTP/1.1
16 20.258173300  172.233.221.124   192.168.126.128   TCP     60 80 → 41234 [ACK] Seq=806 Ack=1021 Win=64240 Len=0
17 20.282050112  172.233.221.124   192.168.126.128   HTTP    4… HTTP/1.1 200 OK  (text/html)
18 20.282077837  192.168.126.128   172.233.221.124   TCP     54 41234 → 80 [ACK] Seq=1021 Ack=1210 Win=31395 Len=0
```

References

mdn web docs, Authorization,
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization
mdn web docs, HTTP authentication,
https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication#basic_authentication_scheme
mdn web docs, HTTP caching,
https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching
HTTP Working Group, HTTP Semantics,
https://httpwg.org/specs/rfc9110.html#field.www-authenticate