Authors: Brendan Banfield
Some work done with Kyra Helmbold but everything in this file is my own
CS338

First, I used nslookup to find the IP of cs338.jeffondich.com, which is 45.79.89.123. Then I set my wireshark filter to ip.addr == 45.79.89.123 and tcp.port == 80, so that I would only see HTTP communication to that website. When loading cs338.jeffondich.com/basicauth, we see a normal TCP handshake followed by a GET request: (Note: TCP Keep-Alive and some handshakes are omitted for readability)

Source	Destination	Protocol	Length	Info
192.168.116.128	45.79.89.123	TCP	74	37090 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4201391040 TSecr=0 WS=128
45.79.89.123	192.168.116.128	TCP	60	80 → 37090 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.116.128	45.79.89.123	TCP	54	37090 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
192.168.116.128	45.79.89.123	HTTP	416	GET /basicauth/ HTTP/1.1

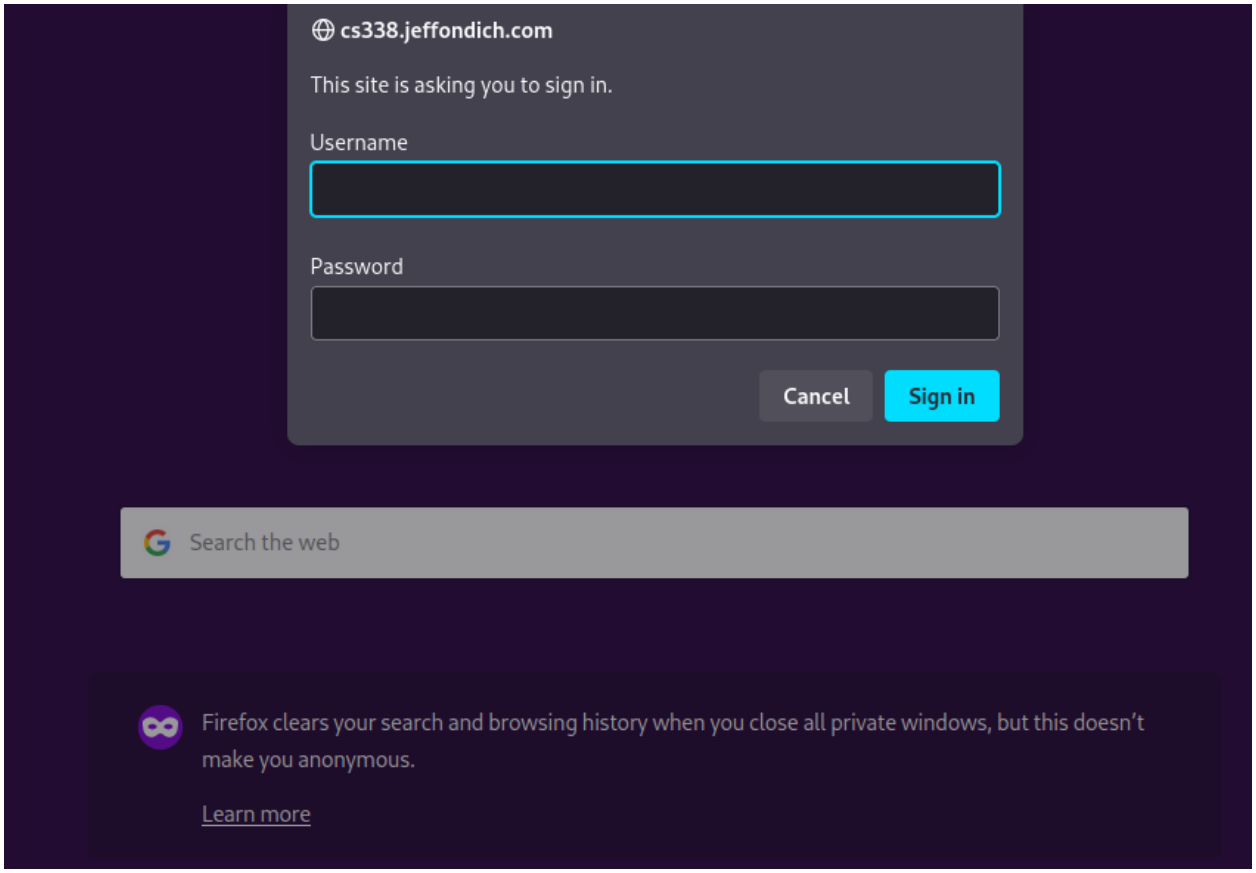
```
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth/]
    [HTTP request 1/1]
    [Response in frame: 230]
```

However, instead of sending the requested resource, the server responds with a 401 Unauthorized:

```
45.79.89.123        192.168.116.128      HTTP      457 HTTP/1.1 401 Unauthorized  (text/html)
```

```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 401 Unauthorized\r\n
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Wed, 20 Sep 2023 17:53:20 GMT\r\n
    Content-Type: text/html\r\n
  ▸ Content-Length: 188\r\n
    Connection: keep-alive\r\n
    WWW-Authenticate: Basic realm="Protected Area"\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.050101690 seconds]
    [Request in frame: 228]
    [Request URI: http://cs338.jeffondich.com/basicauth/]
    File Data: 188 bytes
▾ Line-based text data: text/html (7 lines)
    <html>\r\n
    <head><title>401 Authorization Required</title></head>\r\n
    <body>\r\n
    <center><h1>401 Authorization Required</h1></center>\r\n
    <hr><center>nginx/1.18.0 (Ubuntu)</center>\r\n
    </body>\r\n
    </html>\r\n
```

This has some html attached, though none of it actually displays. Instead, the browser shows a popup window with a username and password prompt on top of the previous page (default opening page in this case)



This is because the 401 contained this tag:
WWW-Authenticate: Basic realm="Protected Area"\r\n
Which requests a basic authentication protocol and names the restricted region. As discussed at https://www.ibm.com/docs/en/cics-ts/5.4?topic=concepts-http-basic-authentication, this requests a username and password from the user. This should be included in the GET request, in an Authorization header. Upon entering the username and password, the browser starts a new TCP handshake and sends sends a new GET request:

```
192.168.116.128      45.79.89.123          HTTP       459 GET /basicauth/ HTTP/1.1
```

```
▾ Hypertext Transfer Protocol
  ▸ GET /basicauth/ HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▸ Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth/]
    [HTTP request 1/2]
    [Response in frame: 573]
    [Next request in frame: 575]
```

This includes the tag
Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
where the string is a base64 encoding of “cs338:password”, the login credentials.
Since these login credentials are correct, the browser now responds with the requested resource:

```
45.79.89.123        192.168.116.128      HTTP      458 HTTP/1.1 200 OK  (text/html)
```

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Server: nginx/1.18.0 (Ubuntu)\r\n
  Date: Wed, 20 Sep 2023 18:01:25 GMT\r\n
  Content-Type: text/html\r\n
  Transfer-Encoding: chunked\r\n
  Connection: keep-alive\r\n
  Content-Encoding: gzip\r\n
  \r\n
  [HTTP response 1/2]
  [Time since request: 0.051152239 seconds]
  [Request in frame: 571]
  [Next request in frame: 575]
  [Next response in frame: 577]
  [Request URI: http://cs338.jeffondich.com/basicauth/]
  HTTP chunked response
  Content-encoded entity body (gzip): 205 bytes -> 509 bytes
  File Data: 509 bytes
Line-based text data: text/html (9 lines)
  <html>\r\n
  <head><title>Index of /basicauth/</title></head>\r\n
  <body>\r\n
  <h1>Index of /basicauth/</h1><hr><pre><a href="../">../</a>\r\n
  <a href="amateurs.txt">amateurs.txt</a>                                    04-Apr-2022 14:10
  <a href="armed-guards.txt">armed-guards.txt</a>                                04-Apr-2022 14:10
  <a href="dancing.txt">dancing.txt</a>                                      04-Apr-2022 14:10
  </pre><hr></body>\r\n
  </html>\r\n
```

If we now click on one of the links (which of course also password protected), another TCP handshake and GET request happen.

```
192.168.116.128     45.79.89.123        TCP       74 60480 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4202190589 TSecr=0 WS=128
45.79.89.123        192.168.116.128     TCP       60 80 → 60480 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.116.128     45.79.89.123        TCP       54 60480 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
192.168.116.128     45.79.89.123        HTTP     520 GET /basicauth/amateurs.txt HTTP/1.1
45.79.89.123        192.168.116.128     TCP       60 80 → 60480 [ACK] Seq=1 Ack=467 Win=64240 Len=0
45.79.89.123        192.168.116.128     HTTP     375 HTTP/1.1 200 OK  (text/plain)
192.168.116.128     45.79.89.123        TCP       54 60480 → 80 [ACK] Seq=467 Ack=322 Win=63919 Len=0
```

The browser automatically includes the authentication information, so the server gives the resource.

```
Hypertext Transfer Protocol
  GET /basicauth/amateurs.txt HTTP/1.1\r\n
  Host: cs338.jeffondich.com\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Referer: http://cs338.jeffondich.com/basicauth/\r\n
  DNT: 1\r\n
  Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
    Credentials: cs338:password
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  \r\n
  [Full request URI: http://cs338.jeffondich.com/basicauth/amateurs.txt]
  [HTTP request 1/1]
  [Response in frame: 762]
```

Importantly, none of this password authentication is encrypted (it's encoded in base64, but encoded != encrypted). Since the connection is only using HTTP, which is insecure, anyone viewing network traffic could see the password exchange (or could just see the resource get sent without even looking at the password).

More details after reading the docs at
https://datatracker.ietf.org/doc/html/rfc7617#section-2
-the browsers request for a username and password is called a "challenge"
-every part of the server with the same realm name has the same seat of usernames and passwords
-Even if sensitive information is not protected, this system should not allow users to enter custom usernames and passwords–they must be supplied by the server. Otherwise, naive users will use a username and password they use elsewhere, potentially compromising their other accounts