

## AUTHOR - KAI JOHNSON

## STORY

### Chapter 1: the handshake

Firstly, the client and server engage in a tcp 3-way handshake.

1. the client (192.168.58.128, the vmware software) sends [SYN] (a synchronization request),
2. the server (45.79.89.123, cs231.jeffondich.com) returns [SYN, ACK] (an acknowledgement and its own synchronization request),
3. the client returns [ACK] (an acknowledgement of the server's message).

10.000...	192.168...	45.79.89...	TCP	74 46102 → 80	[SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1647250750 TS
20.000...	192.168...	45.79.89...	TCP	74 46104 → 80	[SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1647250751 TS
30.043...	45.79.89...	192.168...	TCP	60 80 → 46102	[SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
40.043...	192.168...	45.79.89...	TCP	54 46102 → 80	[ACK] Seq=1 Ack=1 Win=64240 Len=0

### Chapter 2: the HTML request

Now that a connection is set up between the client and server, the client makes its' request: it wants access to the page: <http://cs338.jeffondich.com/basicauth/>

4. the client sends an HTTP GET request for the above page
5. the server acknowledges the request

40.043...	192.168...	45.79.89...	TCP	54 46102 → 80	[ACK] Seq=1 Ack=1 Win=64240 Len=0
50.043...	192.168...	45.79.89...	HTTP	3..	GET /basicauth/ HTTP/1.1
60.044...	45.79.89...	192.168...	TCP	60 80 → 46102	[ACK] Seq=1 Ack=342 Win=64240 Len=0
70.046...	45.79.89...	192.168...	TCP	60 80 → 46104	[SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

### Chapter 3: the initial server rejection

6. the server sends back an HTTP "Unauthorized" packet, with status code 401, telling the client that authorization is required to visit the webpage that they requested,
  - a. issuing a "WWW-Authenticate" challenge for the Basic scheme with a realm value of "Protected Area" (<https://datatracker.ietf.org/doc/html/rfc7235#section-4.2>)

90.088...	45.79.89...	192.168...	HTTP	4..	HTTP/1.1 401 Unauthorized (text/html)
100.088...	192.168...	45.79.89...	TCP	54 46102 → 80	[ACK] Seq=342 Ack=404 Win=63837 Len=0
116.045...	192.168...	45.79.89...	TCP	54 46104 → 80	[FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
126.046...	45.79.89...	192.168...	TCP	60 80 → 46104	[ACK] Seq=1 Ack=2 Win=64240 Len=0

Frame 9: 457 bytes on wire (3656 bits), 457 bytes captured (3656 bits) on interface eth0, id 0  
Ethernet II, Src: VMware\_e0:85:18 (00:50:56:e0:85:18), Dst: VMware\_7c:b6:52 (00:0c:29:7c:b6:52)  
Internet Protocol Version 4, Src: 45.79.89.123, Dst: 192.168.58.128  
Transmission Control Protocol, Src Port: 80, Dst Port: 46102, Seq: 1, Ack: 342, Len: 403  
Hypertext Transfer Protocol  
HTTP/1.1 401 Unauthorized\r\n  
[Expert Info (Chat/Sequence): HTTP/1.1 401 Unauthorized\r\n  
Response Version: HTTP/1.1  
Status Code: 401  
[Status Code Description: Unauthorized]  
Response Phrase: Unauthorized  
Server: nginx/1.18.0 (Ubuntu)\r\n  
Date: Sat, 09 Apr 2022 00:40:29 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 1/3]  
[Time since request: 0.044351173 seconds]  
[Request in frame: 5]  
[Next request in frame: 18]  
[Next response in frame: 20]  
[Request URI: http://cs338.jeffondich.com/basicauth/]  
File Data: 188 bytes  
Line-based text data: text/html (7 lines)

## 7. the client acknowledges the HTTP response from the server

```

90.088... 45.79.89... 192.168... HTTP 4...HTTP/1.1 401 Unauthorized (text/html)
100.088... 192.168... 45.79.89... TCP 54 46102 → 80 [ACK] Seq=342 Ack=404 Win=63837 Len=0
116.045... 192.168... 45.79.89... TCP 54 46104 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
126.046... 45.79.89... 192.168... TCP 60 80 → 46104 [ACK] Seq=1 Ack=2 Win=64239 Len=0
136.116... 45.79.89... 192.168... TCP 60 80 → 46104 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
146.116... 192.168... 45.79.89... TCP 54 46104 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
1510.14... 192.168... 45.79.89... TCP 54 [TCP Keep-Alive] 46102 → 80 [ACK] Seq=341 Ack=404 Win=63837 Len=0
Sequence Number (raw): 149794146
[Next Sequence Number: 342 (relative sequence number)]
Acknowledgment Number: 404 (relative ack number)
Acknowledgment number (raw): 1711240651
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 63837
[Calculated window size: 63837]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x820d [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 9]
[The RTT to ACK the segment was: 0.000018356 seconds]
[RTT: 0.043635008 seconds]

```

## Chapter 4: the extremely secure credential communication

8. once the user (that is, the human trying to access <http://cs338.jeffondich.com/basicauth/> through the firefox browser installed on their virtual machine) submits the credential pair, the client sends another HTTP GET request, with an Authorization header.
  - a. The authorization header indicates the authorization scheme and the credentials.
    - i. As communicated by the server (discussed in 6), the authorization scheme is “Basic.” Consequently, the first portion of the authorization header indicates this authorization scheme: it reads “Authorization: Basic”
    - ii. the next portion are the credentials. The userID and the password are combined to form the user-pass (userID + “:” + password). Thus, a username of “cs338” and password of “password” form the user-pass “cs338:password”. This user-pass is then encoded to an octet sequence, which is encoded by base64. The user-pass of “cs338:password” is encoded as “Y3MzMzg6cGFzc3dvcmQ=”.
      1. encoding from user-pass to octet does not have specific implementation specified in original definition of authentication scheme
      2. for more discussion of base64 encoding scheme, see <https://datatracker.ietf.org/doc/html/rfc4648#section-4>
      3. notably, the credentials are NOT ENCRYPTED; if a program can decode base64 encoding (eg, the wireshark GUI) it can print the sent credentials
    - iii. Thus, the authorization header reads  
 Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=  
 containing the authorization scheme and the credentials.
    - iv. for more on the Basic authorization scheme, see:  
<https://datatracker.ietf.org/doc/html/rfc7617>

```

17.11.10... 45.79.89... 192.168... TCP 60 80 → 46102 [ACK] Seq=404 Ack=342 Win=64240 Len=0
1813.51... 192.168... 45.79.89... HTTP 4..GET /basicauth/ HTTP/1.1
1913.51... 45.79.89... 192.168... TCP 60 80 → 46102 [ACK] Seq=404 Ack=726 Win=64240 Len=0
2013.78... 45.79.89... 192.168... HTTP 4..HTTP/1.1 200 OK (text/html)
2113.78... 192.168... 45.79.89... TCP 54 46102 → 80 [ACK] Seq=726 Ack=808 Win=63837 Len=0

Frame 18: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface eth0, id 0
Ethernet II, Src: VMware_7c:b6:52 (00:0c:29:7c:b6:52), Dst: VMware_e0:85:18 (00:50:56:e0:85:18)
Internet Protocol Version 4, Src: 192.168.58.128, Dst: 45.79.89.123
Transmission Control Protocol, Src Port: 46102, Dst Port: 80, Seq: 342, Ack: 404, Len: 384
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:91.0) Gecko/20100101 Firefox/91.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Authorization: Basic Y3MzMzg6c6Fzc3dvcmQ=\r\n
Credentials: cs338:password
\r\n
[Full request URI: http://cs338.jeffondich.com/basicauth/]
[HTTP request 2/3]
[Prev request in frame: 5]
[Response in frame: 20]
[Next request in frame: 22]

```

9. the server returns a TCP acknowledgement

## Chapter 5: the access

10. the server accepts the credential pair and returns an “OK” package containing the html for the webpage

```

1913.51... 45.79.89... 192.168... TCP 60 80 → 46102 [ACK] Seq=404 Ack=726 Win=64240 Len=0
2013.78... 45.79.89... 192.168... HTTP 4..HTTP/1.1 200 OK (text/html)
2113.78... 192.168... 45.79.89... TCP 54 46102 → 80 [ACK] Seq=726 Ack=808 Win=63837 Len=0

Frame 20: 458 bytes on wire (3664 bits), 458 bytes captured (3664 bits) on interface eth0, id 0
Ethernet II, Src: VMware_e0:85:18 (00:50:56:e0:85:18), Dst: VMware_7c:b6:52 (00:0c:29:7c:b6:52)
Internet Protocol Version 4, Src: 45.79.89.123, Dst: 192.168.58.128
Transmission Control Protocol, Src Port: 80, Dst Port: 46102, Seq: 404, Ack: 726, Len: 404
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Server: nginx/1.18.0 (Ubuntu)\r\n
Date: Sat, 09 Apr 2022 00:40:44 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 2/3]
[Time since request: 0.263589359 seconds]
[Prev request in frame: 5]
[Prev response in frame: 9]
[Request in frame: 18]
[Next request in frame: 22]
[Next response in frame: 24]
[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)

```

11. the client returns a TCP acknowledgement

## Chapter 6: wait, what about that logo?

12. the client also sends an HTTP GET request for <http://cs338.jeffondich.com/favicon.ico>
13. the server acknowledges the request, and returns an HTTP 404 not found response
14. the client acknowledges the HTTP not found message

```

2214.19... 192.168... 45.79.89... HTTP 3..GET /favicon.ico HTTP/1.1
2314.19... 45.79.89... 192.168... TCP 60 80 → 46102 [ACK] Seq=808 Ack=1027 Win=64240 Len=0
2414.28... 45.79.89... 192.168... HTTP 3..HTTP/1.1 404 Not Found (text/html)
2514.28... 192.168... 45.79.89... TCP 54 46102 → 80 [ACK] Seq=1027 Ack=1137 Win=63837 Len=0
2624.47... 192.168... 45.79.89... TCP 54 [TCP Keep-Alive] 46102 → 80 [ACK] Seq=1026 Ack=1137 Win=63837 Len=0

```