Nacho Rodriguez-Cortes & Eric Neidhart

The first step that it goes through is the standard TCP synchronization handshake (although it should be noted that there are actually two handshakes happening in parallel, one to local port 57440 which will be handling actual website content/basic authentication, and the other with local port 57442, which attempts to get access to information involving favicon.ico, which is a way of displaying an icon in the address bar. For the purposes of looking at the basic authentication exchange, we will ignore the packets having to do with port 57442, see the addendum for more information about what we believe is happening with that.). This can be seen in packets 1, 3, and 4 in image 1. Then, we have an HTTP request for the html for the /basicauth/ webpage, that the server then acknowledges (packets 5 & 6). That is followed by the server responding that the user is unauthorized in packet 9, and that HTTP response includes a 401 Unauthorized message along with 'WWW-Authenticate: Basic realm = "Protected Area"' message telling the user that they are attempting to access a site that requires HTTP basic authentication (seen in image 2). The information in this packet is required as part of the HTTP Basic Authentication scheme, with the basic realm string being the identification of the 'protection space', the given example in the official documentation being 'Basic realm="WallyWorld"'(link 2, page 4).

The user acknowledges that message in packet 10, and the browser gives the popup for username and password. Upon entering those, another get request is sent by the user (packet 11), which contains the username and password in the form of a base 64 encoded message Y3MyMzE6cGFzc3dvcmQ (image 3), which, although it is not easily readable by humans, is not any form of actual encryption, as any online decoder can tell you that this is equivalent to cs231:password, aka our username and password to access the site. This matches the documentation in the scheme, which states that in order to

"To receive authorization, the client

1. obtains the user-id and password from the user,
2. constructs the user-pass by concatenating the user-id, a single colon (":") character, and the password,
3. encodes the user-pass into an octet sequence (see below for a discussion of character encoding schemes),
4. and obtains the basic-credentials by encoding this octet sequence using Base64 ([RFC4648], Section 4) into a sequence of US-ASCII characters ([RFC0020]).”

(link 2, page 4)

The server acknowledges that get request in packet 12, and then sends an HTTP packet OK that contains the webpage information itself in packet 13, and receipt is acknowledged by our computer in packet 14. After that, the user is free to navigate to and from the site as they choose, until their browser cache is cleared.

Addendum about favicon.ico:

It is worth noting that the user still completes the TCP handshake with the favicon.ico server port before the unauthorized HTTP message sent by the server in packet 9. It isn't until the user sends a GET request (packet 15), requesting favicon.ico that the server responds with an HTTP 404 'not found' error message (packet 17). At this point, the user acknowledges that error (packet 18), but the webpage itself is not affected since that information was contained in the earlier HTTP packet OK (packet 13). We believe that the favicon.ico is requested through a separate GET request and local port than the one requesting the webpage information because that would allow the webpage to still load the information, even if the favicon.ico is not accessible, as was the case when we attempted to access the site.

References:

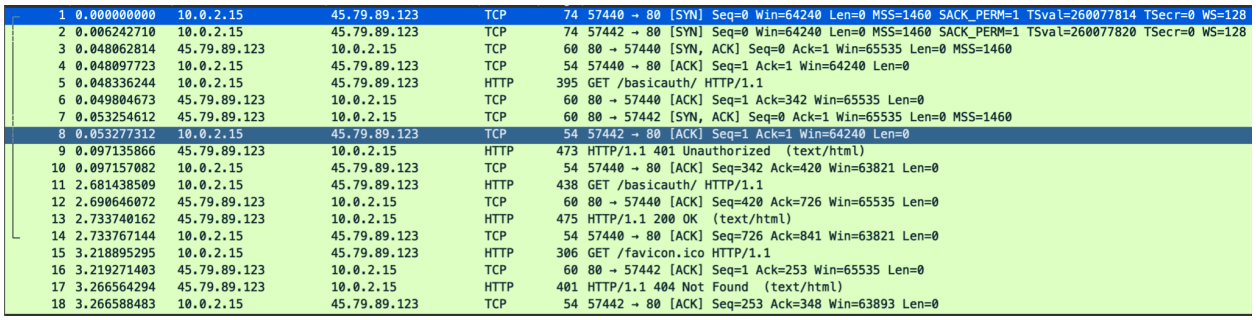
Link 1: our wireshark file:

https://drive.google.com/file/d/1C9QOS2mujLW4efmNVz5Plwj7dJ7obsJ5/view?usp=sharing

Link 2: The ‘Basic’ HTTP Authentication Scheme document

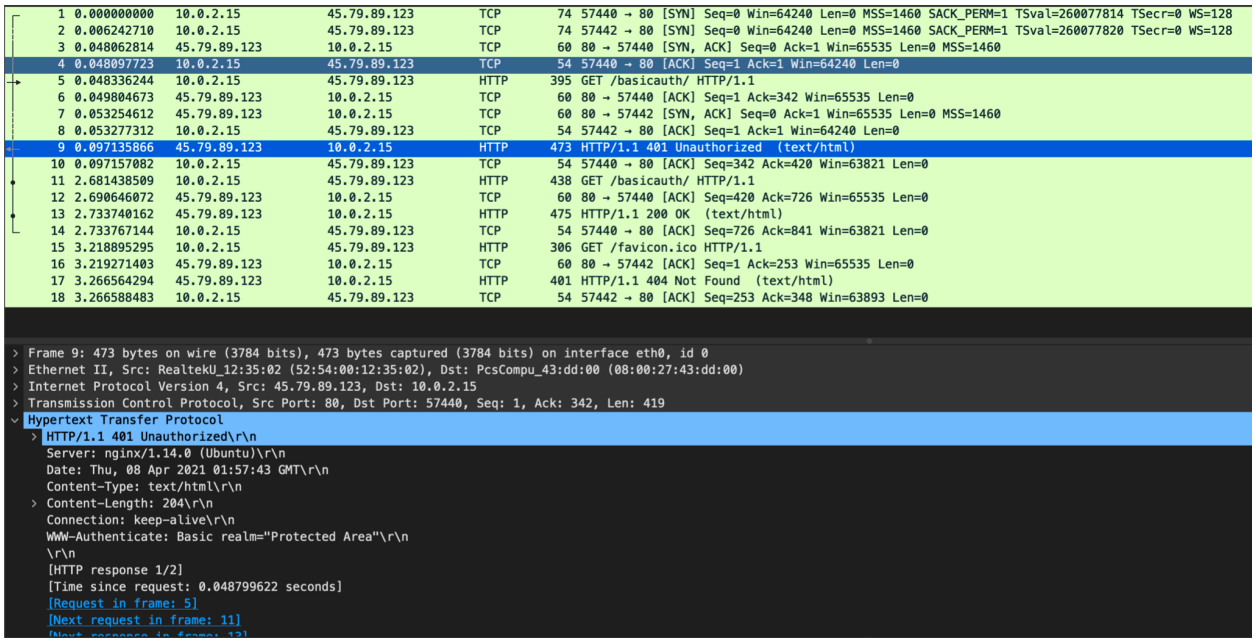
https://tools.ietf.org/html/rfc7617#section-1

Image 1 (summaries of all of our packets):



1	0.000000000	10.0.2.15	45.79.89.123	TCP	74 57440 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=260077814 TSecr=0 WS=128
2	0.006242710	10.0.2.15	45.79.89.123	TCP	74 57442 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=260077820 TSecr=0 WS=128
3	0.048062814	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
4	0.048097723	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
5	0.048336244	10.0.2.15	45.79.89.123	HTTP	395 GET /basicauth/ HTTP/1.1
6	0.049804673	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [ACK] Seq=1 Ack=342 Win=65535 Len=0
7	0.053254612	45.79.89.123	10.0.2.15	TCP	60 80 → 57442 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8	0.053277312	10.0.2.15	45.79.89.123	TCP	54 57442 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
9	0.097135866	45.79.89.123	10.0.2.15	HTTP	473 HTTP/1.1 401 Unauthorized (text/html)
10	0.097157082	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=342 Ack=420 Win=63821 Len=0
11	2.681438509	10.0.2.15	45.79.89.123	HTTP	438 GET /basicauth/ HTTP/1.1
12	2.690646072	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [ACK] Seq=420 Ack=726 Win=65535 Len=0
13	2.733740162	45.79.89.123	10.0.2.15	HTTP	475 HTTP/1.1 200 OK (text/html)
14	2.733767144	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=726 Ack=841 Win=63821 Len=0
15	3.218895295	10.0.2.15	45.79.89.123	HTTP	306 GET /favicon.ico HTTP/1.1
16	3.219271403	45.79.89.123	10.0.2.15	TCP	60 80 → 57442 [ACK] Seq=1 Ack=253 Win=65535 Len=0
17	3.266564294	45.79.89.123	10.0.2.15	HTTP	401 HTTP/1.1 404 Not Found (text/html)
18	3.266588483	10.0.2.15	45.79.89.123	TCP	54 57442 → 80 [ACK] Seq=253 Ack=348 Win=63893 Len=0

Image 2 (401 Unauthorized Packet):



1	0.000000000	10.0.2.15	45.79.89.123	TCP	74 57440 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=260077814 TSecr=0 WS=128
2	0.006242710	10.0.2.15	45.79.89.123	TCP	74 57442 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=260077820 TSecr=0 WS=128
3	0.048062814	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
4	0.048097723	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
5	0.048336244	10.0.2.15	45.79.89.123	HTTP	395 GET /basicauth/ HTTP/1.1
6	0.049804673	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [ACK] Seq=1 Ack=342 Win=65535 Len=0
7	0.053254612	45.79.89.123	10.0.2.15	TCP	60 80 → 57442 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8	0.053277312	10.0.2.15	45.79.89.123	TCP	54 57442 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
9	0.097135866	45.79.89.123	10.0.2.15	HTTP	473 HTTP/1.1 401 Unauthorized (text/html)
10	0.097157082	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=342 Ack=420 Win=63821 Len=0
11	2.681438509	10.0.2.15	45.79.89.123	HTTP	438 GET /basicauth/ HTTP/1.1
12	2.690646072	45.79.89.123	10.0.2.15	TCP	60 80 → 57440 [ACK] Seq=420 Ack=726 Win=65535 Len=0
13	2.733740162	45.79.89.123	10.0.2.15	HTTP	475 HTTP/1.1 200 OK (text/html)
14	2.733767144	10.0.2.15	45.79.89.123	TCP	54 57440 → 80 [ACK] Seq=726 Ack=841 Win=63821 Len=0
15	3.218895295	10.0.2.15	45.79.89.123	HTTP	306 GET /favicon.ico HTTP/1.1
16	3.219271403	45.79.89.123	10.0.2.15	TCP	60 80 → 57442 [ACK] Seq=1 Ack=253 Win=65535 Len=0
17	3.266564294	45.79.89.123	10.0.2.15	HTTP	401 HTTP/1.1 404 Not Found (text/html)
18	3.266588483	10.0.2.15	45.79.89.123	TCP	54 57442 → 80 [ACK] Seq=253 Ack=348 Win=63893 Len=0

```
> Frame 9: 473 bytes on wire (3784 bits), 473 bytes captured (3784 bits) on interface eth0, id 0
> Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_43:dd:00 (08:00:27:43:dd:00)
> Internet Protocol Version 4, Src: 45.79.89.123, Dst: 10.0.2.15
> Transmission Control Protocol, Src Port: 80, Dst Port: 57440, Seq: 1, Ack: 342, Len: 419
v Hypertext Transfer Protocol
  > HTTP/1.1 401 Unauthorized\r\n
    Server: nginx/1.14.0 (Ubuntu)\r\n
    Date: Thu, 08 Apr 2021 01:57:43 GMT\r\n
    Content-Type: text/html\r\n
  > Content-Length: 204\r\n
    Connection: keep-alive\r\n
    WWW-Authenticate: Basic realm="Protected Area"\r\n
    \r\n
    [HTTP response 1/2]
    [Time since request: 0.048799622 seconds]
    [Request in frame: 5]
    [Next request in frame: 11]
```

Image 3 (User get request w/ username and password):