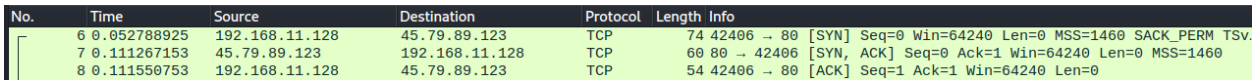
CS338 Basic Authentication: A story

The characters: Browser, Server, User (person)

The goal: To access Jeff's secrets folder via a web browser.

Once upon a time, a user pasted http://cs338.jeffondich.com/basicauth/ into their browser's search bar and pressed enter. The following is the result

The browser starts the interaction by sending a synchronization packet in the hopes of connecting with the server. The server acknowledges that it received the packet and sends one in return. This allows for two-way communication between the browser and the server. The browser then sends another acknowledgment packet back to the server to make it known that they can now talk to each other. This process looks something (exactly) like this:

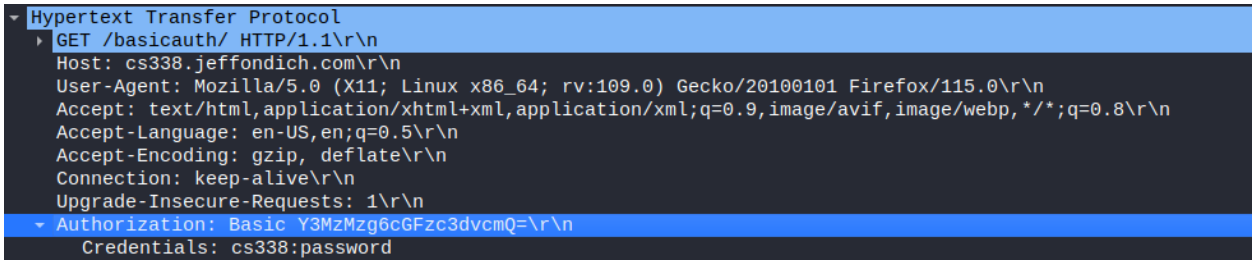


No.	Time	Source	Destination	Protocol	Length	Info
6	0.052788925	192.168.11.128	45.79.89.123	TCP	74	42406 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSv
7	0.111267153	45.79.89.123	192.168.11.128	TCP	60	80 → 42406 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8	0.111550753	192.168.11.128	45.79.89.123	TCP	54	42406 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0

We see the browser (192.168.11.128) and the server (45.79.89.123) setting up the connection, allowing for more fruitful communication in the future.

The browser, now able to speak, asks the server to see the contents of jeffondich.com/basicauth/ (GET /basicauth/ HTTP/1.1). The server acknowledges this request, but, unfortunately for the browser, the server denies this request as the browser has not provided any authorization information. The server does this gracefully by sending a response that reads "401 Unauthorized". The browser then acknowledges that it cannot have access to the site and waits.

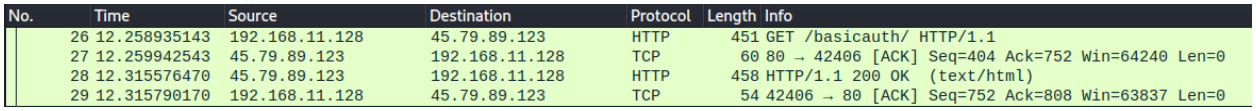
The browser's patience and knowledge pays off. Now that the browser knows what is required to gain access to the site, he prompts the user for the credentials and sends along another request. It is necessary to break this part down into a few parts so that we don't gloss over any important information.



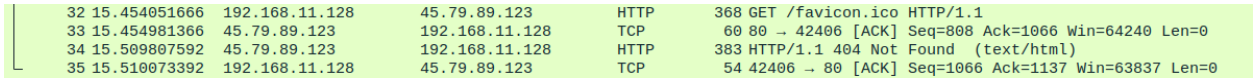
Above is a human readable interpretation of the bytes sent from the browser to the server during its latest GET request.

The authorization header is followed by a seemingly random set of letters and numbers. This is actually the information under the Credentials header! We will get to this in a second. First, what is in the credentials header? Here we find the username (cs338) and the password (password) separated by a colon. This is how the username and password information is formatted before it is encoded. The encoding process (the random string of numbers and letters) requires the splitting of the 8 bit bytes that represent the credentials header into 6 bits that are then encoded into numbers and letters using UTF-8. This is called base64.

This encoding is sent from the browser to the server for authentication. In turn, the server acknowledges the GET request. After a brief moment, the server sends back a 200 OK code along with the website information. This includes the HTML and the secret files. After receiving all of the information that it ever wanted, the browser sends an acknowledgement to the server. All of that together looks like this:



Finally, the browser sends in another request. It asks for the favicon for the website. However, this may have been too greedy. After acknowledging this request, the server searches and searches, but comes up empty handed. As a result, the server sends back 404 Not Found which the browser acknowledges:



Hopefully this brief explanation of accessing Jeff's secret files wasn't too boring or too confusing. All of the procedures outlined above are documented in the HTTP and HTTP Basic Authentication documents if you're interested in more light reading.