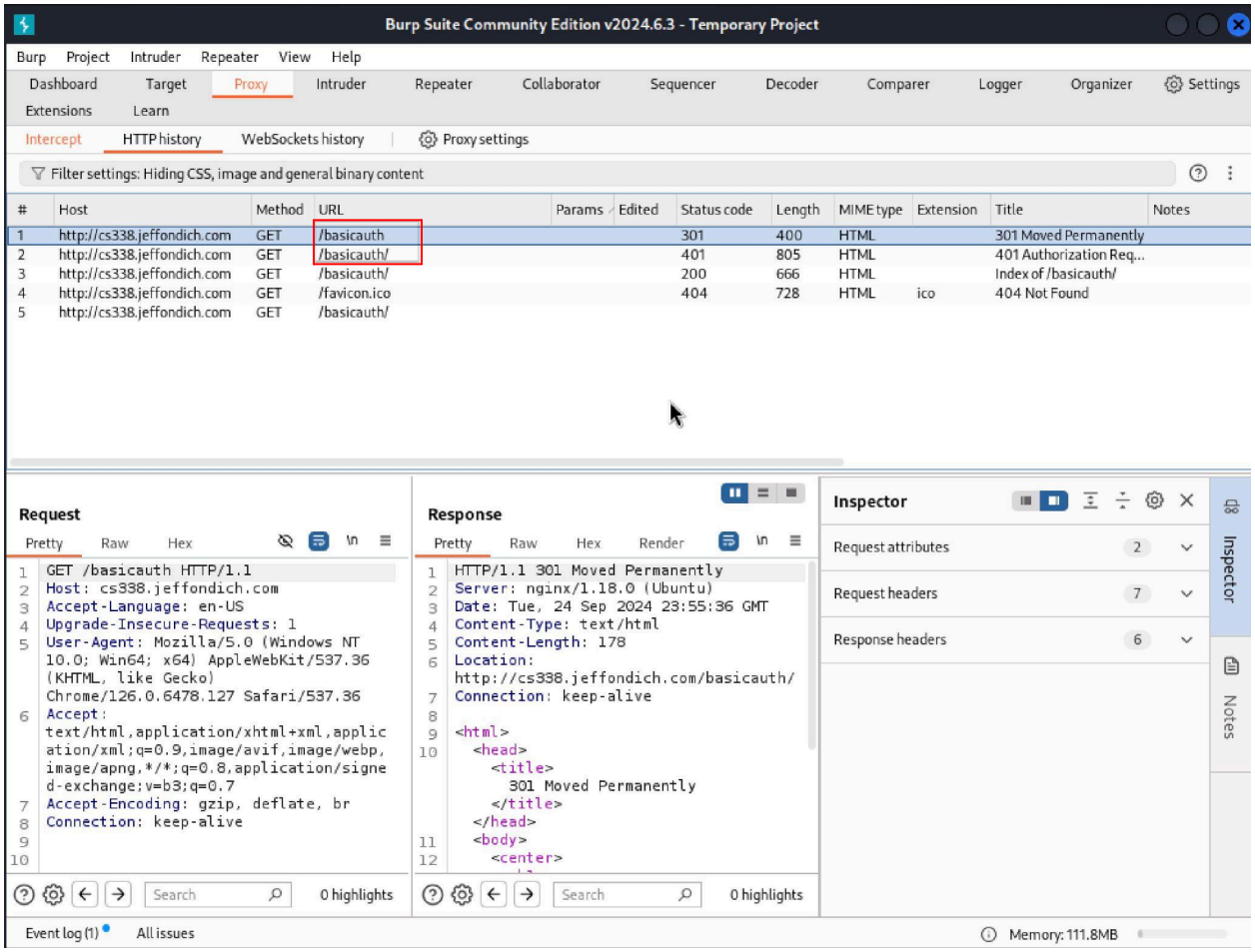
Jeremy Gautama

The Sequence of Events of HTTP's Basic Authentication

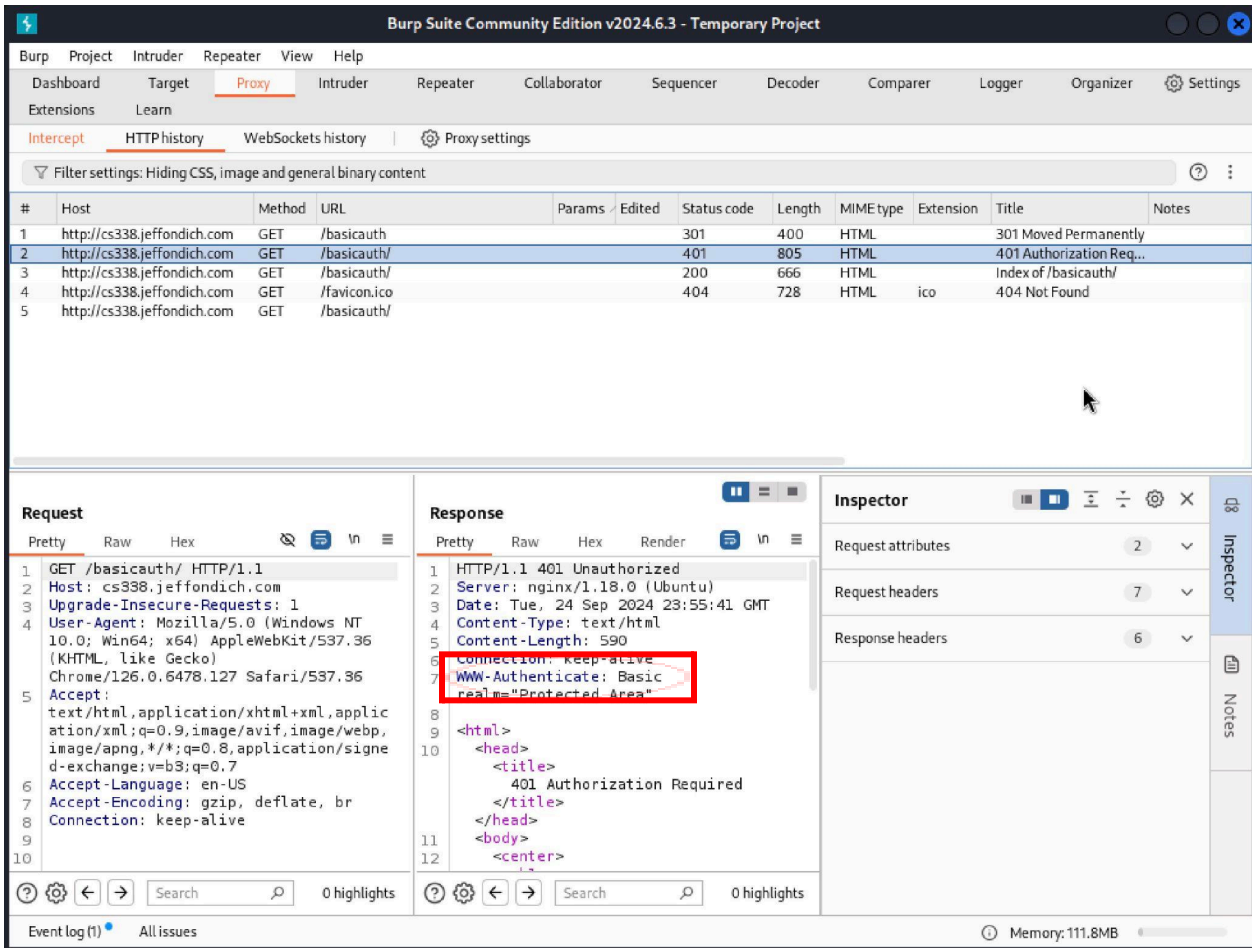
Introduction: I'm using Kali Linux. There's an app in Kali Linux called Burp Suite which freezes per frame to see the HTTP requests and responses, which makes it easier to visualize how these things work.

The Story:
When I first typed in http://cs338.jeffondich.com/basicauth onto Burp Suite's web, my Kali machine's browser requested it using the `GET HTTP/1.1 protocol`. The host server (Jeff's website) responded with a message saying `301 Moved Permanently` – which basically can mean that the original page URL was changed. If you see the `URL` title (header's row, fourth column) in the table of the image below, the URL changed from `/basicauth` to `/basicauth/`. There's an additional backslash redirect that happened. In simple terms, it's just a good samaritan who tells me that the office is here and not there, even though both office doors look so identical.



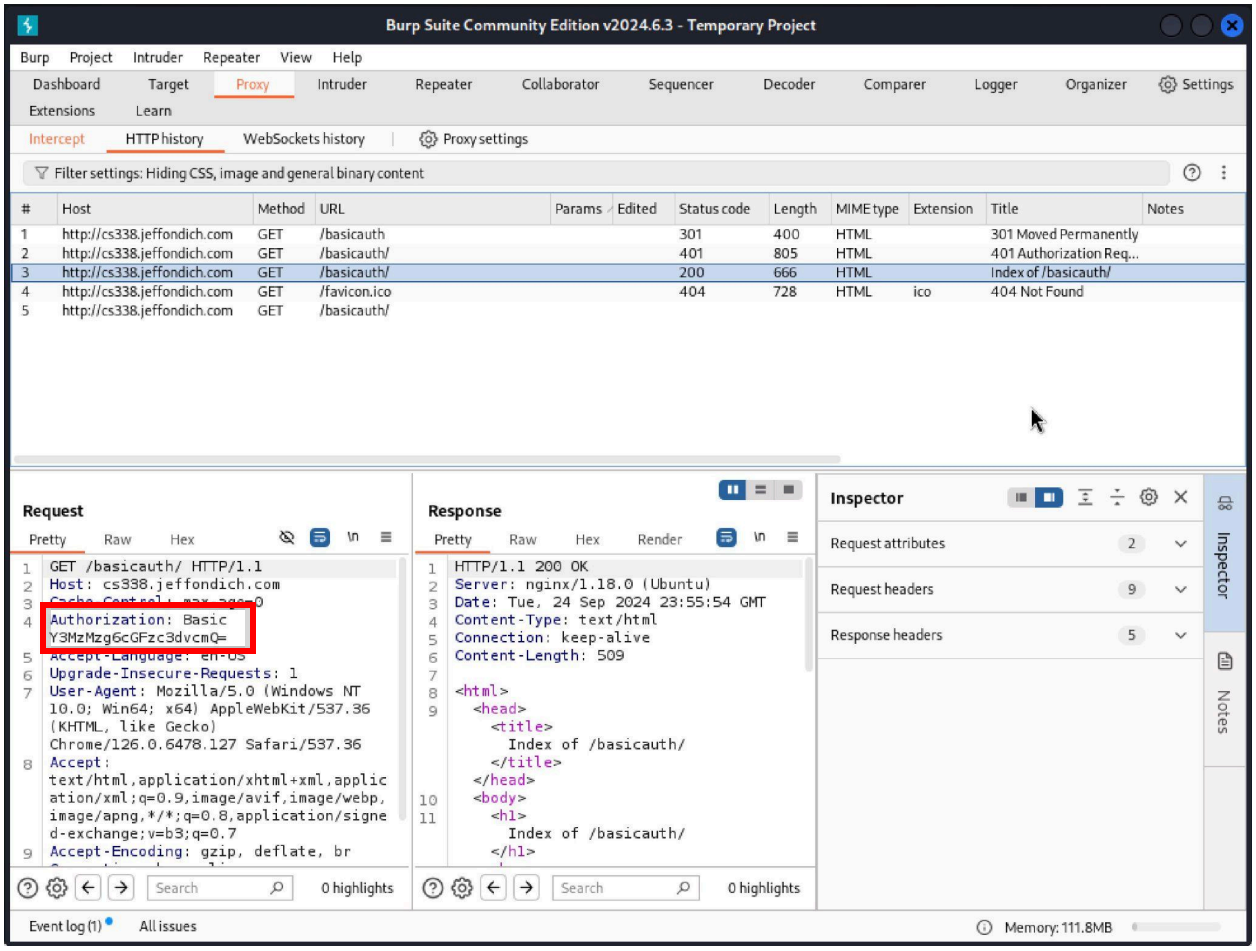
Jeremy Gautama

But then security came along and stopped my request from entering `/basicauth/`.



The security responded with `401 Unauthorized` from my original request, and they also have an additional `WWW-Authenticate` header, basically asking my ID (in this case username and password) if I'm the right person to enter.

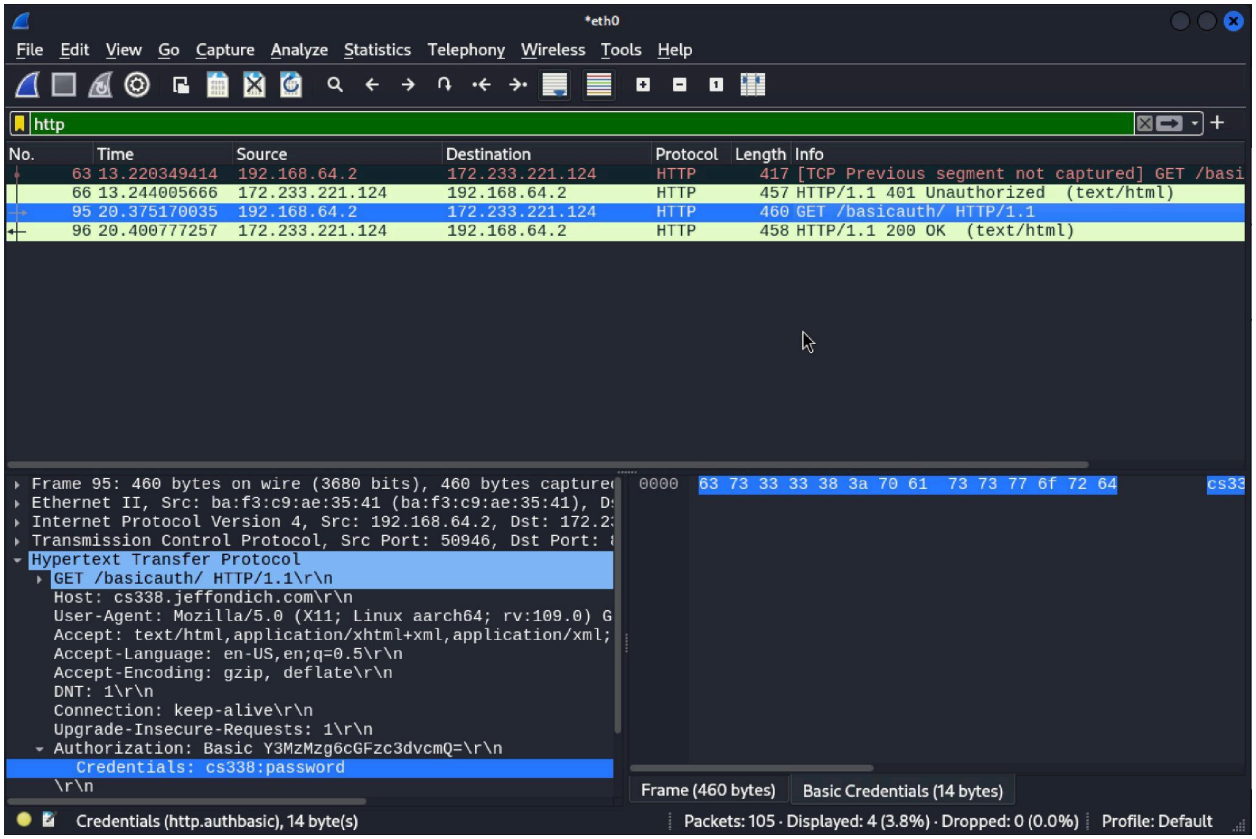
Jeremy Gautama



So I gave my ID by typing in the username `cs338` and password `password`, and with that information my browser sends another `GET HTTP` request to the server with an additional `Authorization` header containing the base64 encoded username and password, which is in the red square box on the image above. The security was happy and said `200 OK`, meaning that I'm authorized to enter.

But the problem here is that maybe the good samaritan was eavesdropping when I said the username and password.

Jeremy Gautama



And if the good samaritan actually did eavesdrop, he basically knows my ID, both username and password by using WireShark. This is because the website was only using HTTP and not HTTPS.

MORAL OF THE STORY: Always be cautious when you do interactions that require usernames and passwords, because there may be a security breach when you input something in a website that is not secure.

So What is Authorization Header?

In the words of Mozilla, “The HTTP Authorization request header can be used to provide credentials that authenticate a user agent with a server, allowing access to a protected resource.” In the case of HTTP Basic Authentication, the header has the base64 encryption for the username and password. If you see from the WireShark screenshot, it is formatted with `Authorization: Basic` and the base64 encryption, and the unencrypted username and password underneath is separated by a colon.
What this basically means in terms of the story is that I give the security guy my ID, he authenticates me, and if I’m on the list of authorized users, I’ll be let into the office.

Jeremy Gautama

Sources:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/301
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization