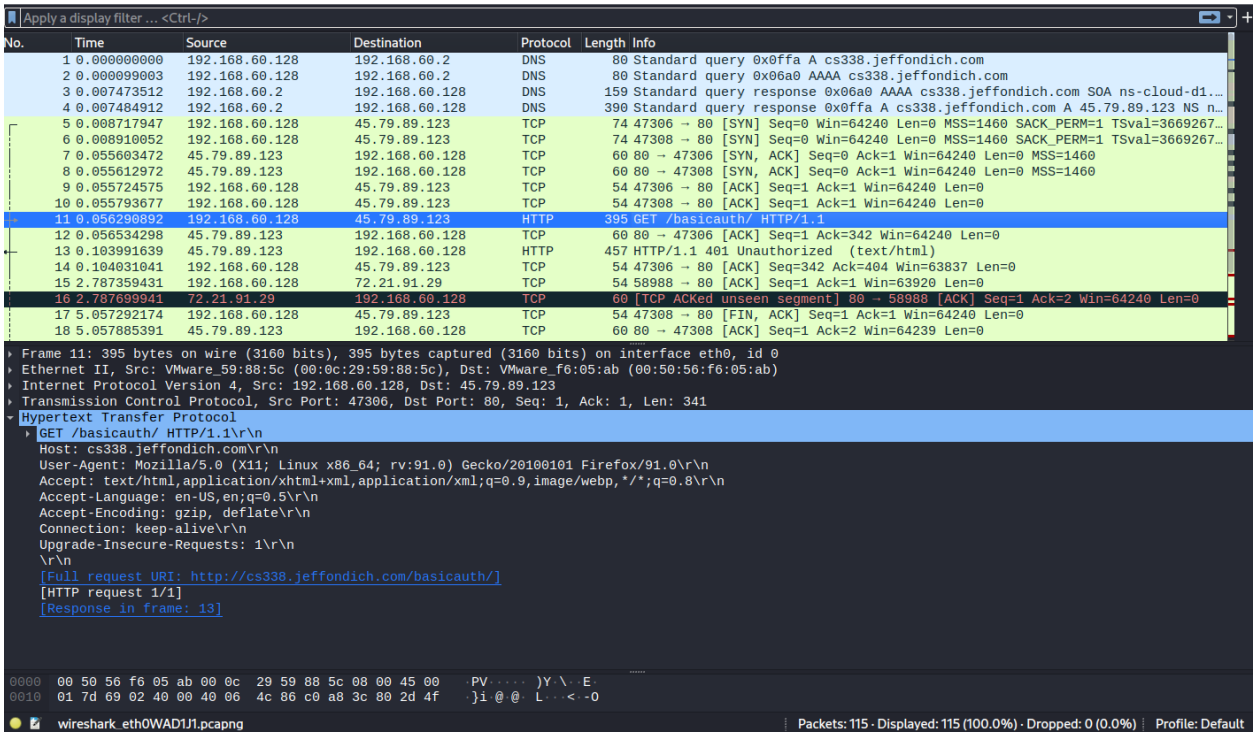
Silas Rhyneer and Charlie Roslansky

We have some general notes on what happened and what we learned at the top, but wrote a more extensive walk through with screenshots in the pages below.

General Notes & Sources

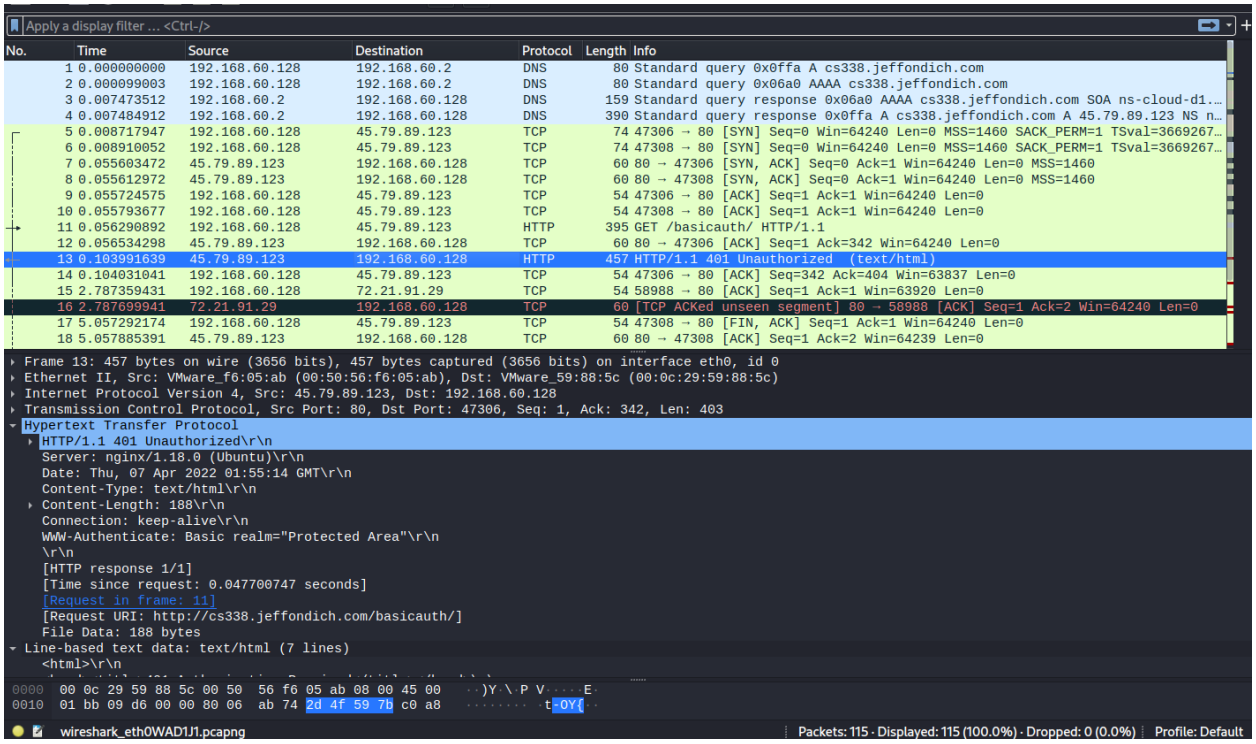
- 2 ports that did the handshake, only one sent a GET request, other closed connection immediately
- PSH is like the elevator button—pushing the current message to be immediately processed
- Sequence num becomes acknowledgement num
- First GET request has no authorization, server responds with WWW authorization header
- ARP translates between IP and MAC addresses
- Future GET requests to home page do not make HTTP requests, page is cached
- DNS lookup, A handles IPv4, AAAA handles IPv6 - only A returned an address
- ~600 DNS servers in the world
- We make request for favicon.ico from different port. It does not exist, and returns 404. Connection closes. We reset connection, and don't try to access it any more
- We try to connect from multiple ports sometimes, and when they fail, rather than retrying on the same port immediately, we just switch to another port and reconnect from there
- It seems that using multiple ports potentially speeds up communication
- Resets whenever it serves the resource
- Going back just retrieves cached html from browser
- Has to make new connection whenever we try to connect to a new link
- When we're not doing anything, it will make Keep-alive calls
- A lot of other stuff unrelated to what we are specifically doing happens in the background
- https://datatracker.ietf.org/doc/html/rfc7617#section-4



In this screenshot, several things are happening. First, look at the lines highlighted in blue (frames 1-4). The first two are requests to a DNS server for the IP address of the domain we just searched. There are two, because one is A type and the other is AAAA type—standing for IPv4 and IPv6 respectively. We get a response back from both, and the A type one gives us an IP address (the one used throughout the rest of the process).

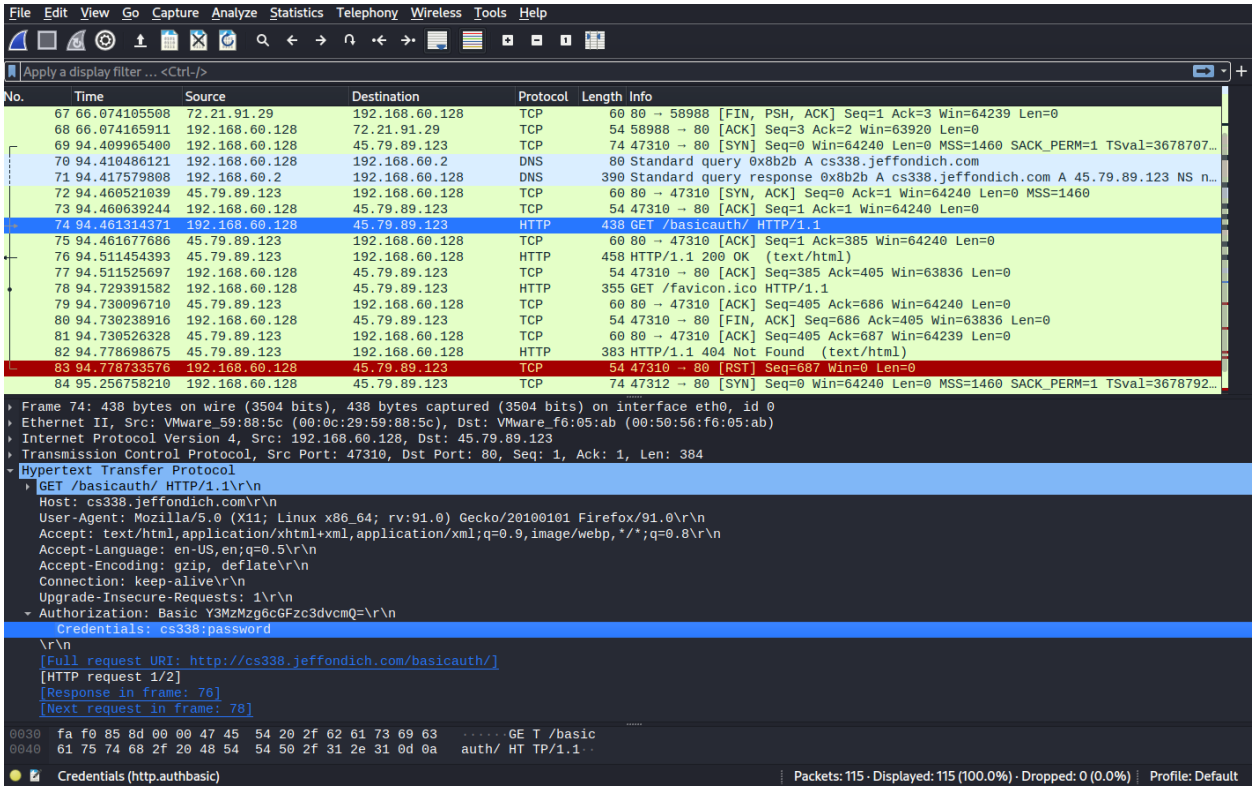
Next, we connect to the server. Frames 5-10 are the three-way handshake. There are 6 messages because we try to connect from two different ports on our computer—both port 47306 and 47308. From what we can infer, this is to accelerate loading times on our computer, since we can have multiple channels open at once.

After making both connections, the first port to finish makes a GET request for /basicauth/. This fails, since it requires authentication.



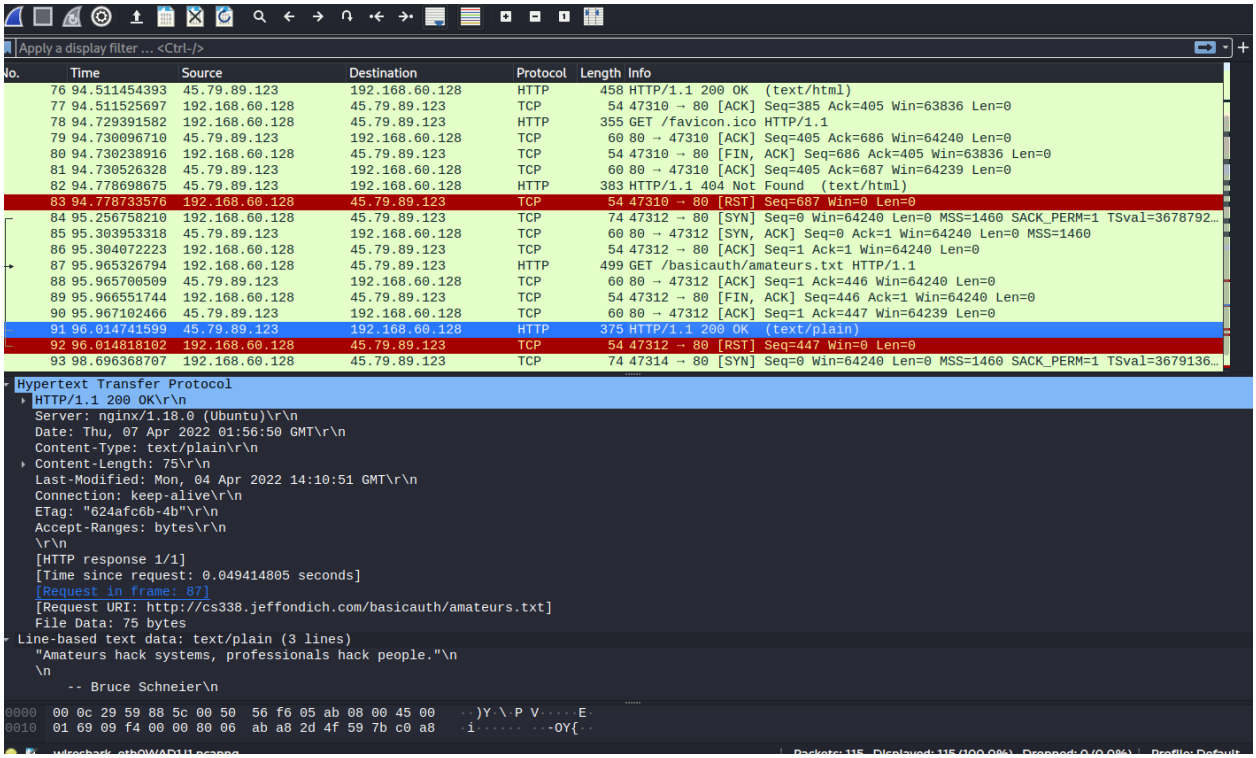
This request fails, and sends back a 401 error, and gives us some html text telling us we can't access the requested files.

Note that in our response from our server, there is a HTTP header called WWW-Authenticate. This means that access to files in this region require username and password. After the header, it says that the authentication scheme is "Basic" and that this realm is "Protected Area". The "Basic" scheme tells the client how to format its authentication, and the region is just the portion of services protected by this authentication. If the region changed, we'd need to reauthenticate.

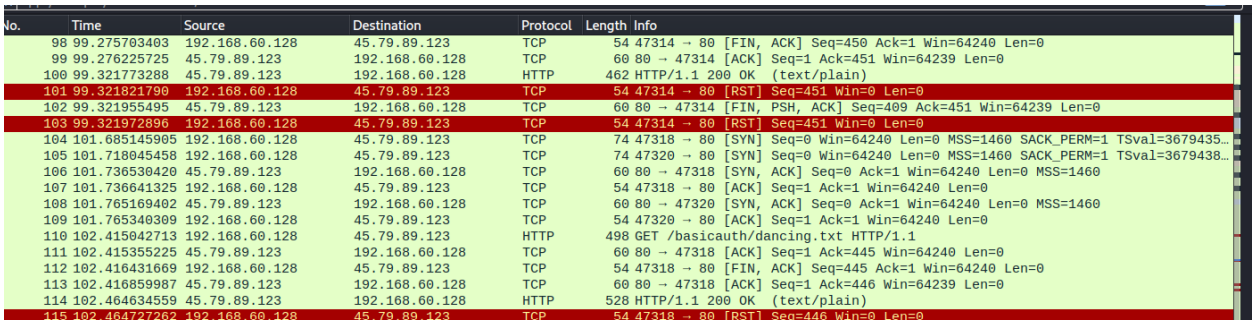


Upon hitting submit on our authentication, we are allowed back in. Once again, we make a request, this time passing in our HTTP with a new header: “Authorization”. This header includes our credentials, which are converted to base64 and are not encrypted. The server then verifies them, and grants us access if our credentials match.

After gaining access, we can now load the page. The server sends back the page with an OK message, and then attempts to fetch the icon for the tab. This request is made by the other port that we opened up, and we get a 404 page not found error, since there is no icon that exists for the page. We also send a reset (RST) message to the server, essentially asking to start over, since we’ve gotten what we’ve requested.



After this, we can click on the pages linked in the doc. We have to reconnect, because of the RST, so we reconnect on a new port, and request the page. Since we're already authorized, we don't have to reauthorize ourselves. This is probably because we have been temporarily saved as a "trusted/authorized client" on the server.



Future requests are as simple as making new connections, http requests, getting served the page, and then finishing our connection (FIN), and resetting. Note that when we go back to the previous page, it does not make a new request at all. This is because the previous page was cached entirely by our browser, and therefore *it* can serve us the previous page, rather than requesting it from the server. Sometimes multiple ports connect, sometimes not, though this seems to be largely random, based on our repeated tests.