Zach Weske, CS 591

Describe the various steps involved when you enter a URL in the browser. Make sure to include all the different services that will be used to fetch the page.

DNS Lookup

- First, the browser will attempt to resolve the domain name to an IP address. It will first ask the local computer to see if the computer has it in the hosts file or it is cached. If it is not, the computer will make a request to the TLD authoritative name server (edu). This DNS server will respond with the address of the ".edu" DNS server. Then the request for ua.edu will be sent to the .edu server. Lastly, a request for cs.ua.edu will be sent to the ua.edu authoritative name server. The final result is the IP address of the cs.ua.edu server.

Initiates TCP (Transmission Control Protocol) connection

- In order to initiate an web request, the browser sends a TCP SYN packets. This serves as the initial hello to the server. Then the server responds with a SYN/ACK packets, indicating that it acknowledges the hello. Lastly, the client sends an ACK packet which completes the initialization of the connection. This is called a TCP handshake.

Initiates TLS (Transport Layer Security)

- The client sends a hello to the server. The server replies with a random string and its public key infrastructure (PKI) certificate, which allows the client to verify that the server is who it says it is, verified by a chain of trust going to a root certificate authority which has trust baked into the operating system. Next, the client sends back a suite of cryptographic ciphers that it can use as well as some other parameters while generating a pre-master secret. Lastly, the server decrypts the pre-master secret and uses it, the random string of bytes, and the cryptographic algorithms from the proposed cipher suite to establish a secure and fully secure symmetrically encrypted communication channel.

Sends HTTP(S) request

- Finally, the browser can send an HTTP request to the server. This details version numbers, the request method (likely a GET request), the resource involved, and any additional information used such as login cookies.

Server sends response and client receives page

- The server sends the HTTP response. This will include version information again as well as the information included in the webpage. As the base HTML/PHP page references images, CSS, JavaScript, or any other files, the browser makes additional requests for the additional resources.
- Lastly, the client renders the page on the screen based on the content type and inserts all additional requested resources. It can also cache data locally to reduce traffic in the future, which is resolved during the GET request for each file.

Feedback Questions (answer to these questions has no impact on your grade)

1) **Was this homework too difficult, or too easy?**

   This homework was easy to implement, though I spent a lot of time on it initially as I was trying to get it to work with a different dataset, which added a bit of confusion on my part.

2) **Was the assignment fun or challenging?**

   The assignment was fun. I have written web servers before but never in a way that used a tool such as this.

3) **Was there something that was unclear?**

   Not really.

4) **Was the homework too long for the given amount of time?**

   It was fine.

5) **What did you learn from this homework?**

   I learned about how Django handles API requests as well as how it stores its data by default.

Sources:

https://realpython.com/api-integration-in-python/#django-rest-framework
https://aws.amazon.com/blogs/mobile/what-happens-when-you-type-a-url-into-your-browser
https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/