SER321 Todd Martin
Assignment 2 : Upper Layers tcmart14

Task 1

Screen shots and packet capture in folder.

1. Filter used was "tcp.port == 465"

2. Standard port for SMTP is either port 25, 465, or 587. However, unlike port 25 and port 587, port 465 was never registered as official in a specification. Instead, the IANA designated that port for SMTP use. It was done so to group in it with other ports that use a secure socket layer, encrypting traffic over the internet. “helo gmail.com” is a command sent to the sever to identify itself. “mail form:” and “rcpt to” identifies who the email is coming from and who it is to.

3. The first command with “openssl” is a command line tool for invoking ssl in encryption. The flag s_client means to open a generic SSL/TLS client. Connect is a flag that designates that the next text is the server being connected to. “auth login” authorizes gcloud access to the cloud platform with user credentials. “data” signifies the beginning of where data is entered, in this example with a subject line being first. The period at the end shows it is the end of that current data stream.

4.I see around 34 lines in wireshark, which it looks like it back and forth communication for establishing a connection using TLS. Around 39 lines in wireshark of frames until the network communication is comprised strictly of application data over TLS.

5. The port of my computer being used is 50435

6. The server sends the first FIN flag. Then the client acknowledges. The client also sends one last packet using TLS then send a FIN ACK to the server.

Task 2

1.

memoranda/commits?branch=default&until=now
This API call sill bring up all commits from the default branch of the memoranda repository up until now.

memoranda/commits?branch=default&until=now&per_page=100
Will return in a similar fashion as the previous call, but the previous call defaults to displaying only about 30 commits. This one expands it to 100 commits.

2.

First I had to do some googling about making API calls in the browser to understand the a question mark signifies the beginning of a query and how to apply logic through ‘&.’ After

that, I looked through the github api and read what all calls in API existed along with filtering methods that applied to the task.

2. A stateless protocol is when a client and server are communicating and when the client sends a request, the server responds in accordance with the state. It also does not require the retention of session information or status of each client-server pair.
   In a stateful protocol, when the client send information, if it does not get a response, it will resend the request. Servers in a stateful protocol also keep information about the connection information and it depends on the state of the server.

Task 3.1

No deliverable

Task 3.2

Screenshots in directory

Task 3.3

1. The filter I used was tcp.port == 9000 since the webserver runs on port 9000, traffic between the browser and server will include packet containing that port number. This will to capturing traffic in wireshark that is happening between server and client.
2. Hitting random versus loading the page, I do not see. Difference in the web traffic.
3. When the browser refresh is hit, we get "Received: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8" where as hitting the button on the page we get "Received: Accept: */*" in the terminal output. It appears that the browser refresh regenerates the page completely, while the button on the page itself just generates new text and image without reloading the whole page.
4. I am able to get 200, for successful response to a request. Also, a 400 response code for a bad request. 404 request for file not found when I enter file/sample.htm
5. I am able to find all the data that the server sends back when filtering for traffic on port 9000. For responses, I get 200 responses for webpages loading up without error. Some pages return a 400 where the request was bad. I also get 404 for file not found.
6. HTTPS provide the use of “Secure Socket Layer” that encrypts the data between the client and server. This is becoming more common as HTTP allows for traffic and data to go across the wires as plain text and is vulnerable to someone who is doing something as simple as running wireshark to capture packets.
7. The server uses port 9000. This not the most common port for HTTP as the standard port for HTTP is port 80.
8. Port 51326 is used on my local computer. There are dynamic ports that are not reserved and can be used by the system as need.

Task 3.4

1. The traffic is going through port 80
2. It is still HTTP since it does not have a certificate and configuration done to ensure SSL.
3. I do not see a difference in the URL