For my final project, I wanted to make a program that would allow me to retrieve my stats from the video game Rocket League. I intended to decrypt the data sent to the Rocket League servers using a client-side attack. I then would write a python program that would retrieve the decrypted Application Data written to a text file and format it to my liking. After everything I know, the task seems pretty easy to complete. I ran into the problem that the Application Data is already encrypted with Epic Games' private key before sending it over the wire. To decrypt the data, I would need to steal the private key from Epic Games, and for legal reasons, unfortunately, I am unable to complete my project. If I had the private key, the project would be as simple as reading the data into a text file and then using python to write at most 30 lines of code to format the data.

I originally started out by using Wireshark on my virtual machine. When I opened up Wireshark, it was not capturing any packets, so I went to the network setting for my virtual machine and changed it from a NAT Network to a Bridged Adapter. That allowed me to capture packets, but for some reason, it didn’t seem like it was capturing the correct packets. So I decided to scrap the virtual machine, and I downloaded Wireshark onto Windows 11. When I loaded Wireshark up, the packet capture started to work perfectly.

I don’t know what I was capturing on my virtual machine, but for whatever reason, the packets looked way different from the packets I was capturing when using Wireshark on my device. I then loaded up Rocket League and played a few games just to see what the packets looked like and if I could see any patterns arise. I was particularly looking at the end of each game to see what was sent to Epic Games after the game ended. I saw a lot of protocol TLSv1.2 packets being sent with the info section saying Application Data. I realized that this is what I needed to decrypt to find the information I was looking for. I went to Youtube to watch a few videos to get a feel for how I am supposed to decrypt this application data. The first video I watched was the most helpful in decrypting any data. To decrypt the data, I needed to capture the TLS keys. I went to the control panel, advanced system settings, environment variables, and set up an SSLKEYFILE (SSL is the old version of TLS) that would log the TLS keys in a text file. This was pretty easy, and it logs all of the keys from every internet search into that file. To decrypt the data, you need to capture the packets on Wireshark while logging the TLS keys. A weird, unnecessary obstacle that I faced was that Wireshark would not decrypt the packets if I imported the log file after I was done capturing packets. I had to give Wireshark the file path before starting the packet captures. I say this was unnecessary because I watched "How to DECRYPT HTTPS Traffic with Wireshark" in the Youtube video, they were able to give the file path after the packet capture, and it worked perfectly. In another video, "Decrypt TLS traffic on the client-side with Wireshark," they gave the file before the packet capture. I tried that, and it worked. I was able to fully decrypt the HTTPS traffic. With that, I can mainly steal usernames and passwords from a user when they visit any website that uses HTTP/S encryption which is all websites.

To fully understand why I could only decrypt the HTTPS traffic and none of the Application Data. I needed to watch "How TCP really works // Three-way handshake // TCP/IP Deep Dive" and "Hacking the TLS Handshake and decryption with Wireshark // SSL Deep Dive." Those two videos by David Bombal are both an hour each, and I learned so much information from them. The first video goes over everything TCP, which helped me understand TCP protocol while understanding the window size and the flags. The second video went over TLSv1.2 decryption and showed me how to decrypt a TLSv1.2 protocol with a private key and gave me the knowledge of how private keys and public keys are made. This is also the video where I realized that my project wouldn’t be possible to complete without Epic Games' private key.

I overall learned a lot about how encryption, decryption, TCP, and TLS work. At a medium level of expertise, I now understand how they all work independently and work together. I have also gotten comfortable with Wireshark, which is a good skill for me to have since I am looking at going into penetration testing. Before this project, if you had asked me anything about Wireshark or TLS, I wouldn’t have been able to tell you anything, and now I have an excellent knowledge of both.