CSCI 4220 Lab 9

Lab 9: HTTP/2 TLS and QUIC Wireshark

In this lab, you will use wireshark to capture http traffic to understand HTTP/2 handshake we explained in class and also the quic protocol.

For part 1, check the page 9 in wireshark-tls-debugging pdf file and see how to capture TLS session secrets into a key log file. In this part, we use `curl` command to request the google search page, get TLS session secrets. At the same time, you start traffic capturing in wireshark before running `curl` so the http traffic of curl will be captured. In Macos, Ubuntu or linux/WSL, make sure curl 8.4.0 or higher installed. It is better not to use web browswer to visit the google page or service during the following steps.

1. export SSLKEYLOGFILE="$PWD/part1keys.txt" (you can replace it with your own path)
2. curl https://www.google.com (start wireshark capture before running it)
3. After curl returns the content, stop capturing in wireshark.
4. check keys.txt. If you want to understand the format, check https://udn.realityripple.com/docs/Mozilla/Projects/NSS/Key_Log_Format
5. Use the filter of `ip.host==www.google.com` to get the curl http traffic (if it does not work, go to wireshark preferences->Name Resolution-> check Resolve network(IP) addresses)
6. Export the packets File->Export Specified Packets->Export as pcapng, and select `Displayed` and `All Packets`. Use part1 as File name.

Now open the part1 file (the first packet should have value 1 at No. column) and analyze the captured packets in part1.pcapng and answer the following questions. Submit your answers in par1.pdf file along with keys.txt and pcapng file.

1. list the essential packets No. used for TCP/TLS1.3 handshakes and write down their message types. There should be total 5 or 6 packets.

2. Pick an application data packet and write down its No. What's the payload you can find in that packet?
3. Now it is the time to use the keys to decrypt the traffic. go to Preferences->Protocols->TLS->(Pre)-Master_scret log file. Select the part1keys.txt you generated before. Compare with the payload you saw before in your selected application data packet , what's the difference?
4. Continue to find a packet No. that contains plain text HTTP header method :GET. Write down the packet No. and all other header fields and values in the packet.

For part 2, we want to examine Http/3 QUIC traffic. However, http3 is still experimental option for curl. You have to use firefox or chrome to access google page. Check page 8 in wireshark-tls-debugging pdf file and see how to capture TLS session secrets with these apps. If you are using macos, you may have to change the app name with its path name like `/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome` . so the command will be something like
`SSLKEYLOGFILE="$PWD/part2keys.txt" /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome www.google.com --user-data-dir=/tmp/cr1`

Similar to part 1, start capturing before running the command so wireshark can capture the web browser's http traffic. After the webbrowser shows the google page, close it, and stop capturing.

1. check part2keys.txt to see if the session secrets are captured.

2. use the filter of “ip.host==www.google.com and quic” to save the packets in the same way and use part2 as File name.

Now open the part2 file and answer the following questions:

1. List the No. of packets that contains TLS hello messages, and describe how you find them.
2. List the all the frame types in the QUIC packet that contains clienthello message.
3. Do you see any stream frame types in payload packets? If you see it, list packet No. If not, explain why.
4. Now repeat the same decryption step by using part2keys.txt. Look for any packet that contains plain text HTTP header. If you find one, write down the Packet No. If not, explain why.

Submission

Submit a single PDF, Lab9_answers.pdf that contains your responses to all the questions seperated for part 1 and part 2, and upload part1 part2 keys files and pcapng files .