Jerry Lau

CS325 – Final project

Deliverables

The module which I implemented was a bit convoluted as it required going from packet capture using WireShark then pushing that data to python which would do the clean up. The scapy module is also an additional SSL/TLS library which is called upon separately to the regular scapy module used to sniff/spoof packets. Then we parse the data into a c file which will generate the output data file which is human readable along with the graphs for the TLS handshake. These are the points 1-2 of the midterm project proposal which was presented.

Lastly, a session hijacking module was attempted as there had to be an attack on the network of some sort. I selected TeamViewer to be attacked as it was a fairly simple TLS chain along with it's support to various operating systems. The company also claims that the connection is encrypted so that was also proved. As mentioned in the presentation, because TeamViewer is a concurrent program, many packets are sent even if nothing happens between the 2 clients. So in order to actually get the correct sequence number and acknowledgement numbers I did a simple random value to be appended to the most recent packet, as such it was not successful.

Final thoughts

This project helped me better understand the process and method which the TLS handshake is completed. The overall process is tried and true so there is not a lot of variation on it. The way which applications use it vary as some have cached credentials or sessions which allows it to reuse the preordered method. While others like TeamViewer is not cached and as such will always follow the same process.

There is also a lot of additional ways which data mining and machine learning can help us understand and build a more resistant networks by capturing data packets. It is of course just time and computing resources which have to be used for analysis and creation of potentially better networks and networks attached devices.