– Project 5 –
Network Flow Anomaly Detection
Advanced Network Security

Mohammad Umeer – #4748549

Task 1

To monitor the TLS traffic I developed a system composed of two elements, the first one is called "python_data_reader.py" and its role is to read the input *.pcap file and using scapy (custom library for TLS/SSL) parse the content and save it to an output text file called "rawData.txt", unfortunately, the scapy parsing is not perfect because of few handshakes are always misparsed (as you will see in Task 2). I tried different solutions, but I could not found something more reliable than this one.

In the second phase a program called "SSLStudy-UmeerM.c" read and analyse the data of the first part, a classifier algorithm is able to distinguish the handshake type form all the packets. The classifier also generates a file called "logDataOutput.txt" in which it stores all the detected handshake and the relative sockets. The classified data is now analysed using Makarov Chain and the output of this elaboration is visible as a table/s inside the file"tableDataOutput.txt" and as graph/s in a file called "imageGraph.ps"

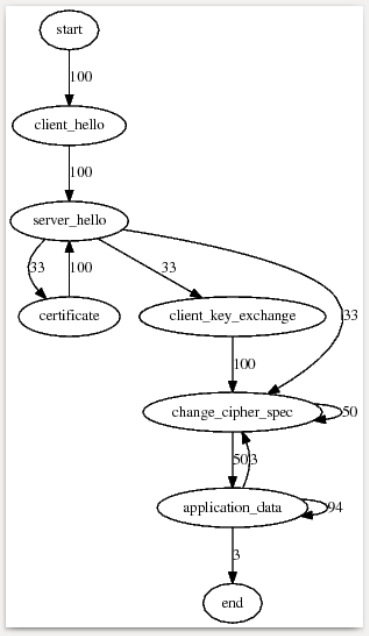


Figure 1: Output graph of the "testing.pcap" file

How to test it:

1) Run the script run.sh by default the system will work with "*testing.pcap"* file but you can also specify a pcap file just by passing the it's location as a parameter.

2) There are three output files:

- ImageGraph.ps = Graph/s of the Markov Chain
- logDataOutput = All the detected handshake with its communication sockets info
- tableDataOutput = Table/s used In the elaboration phase

Source: https://github.com/tintinweb/scapy-ssl_tls

Task 2

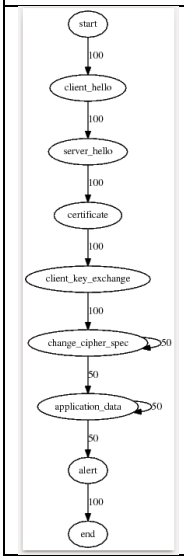
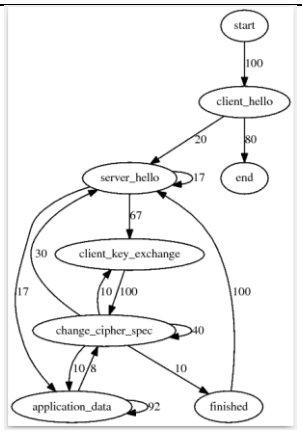
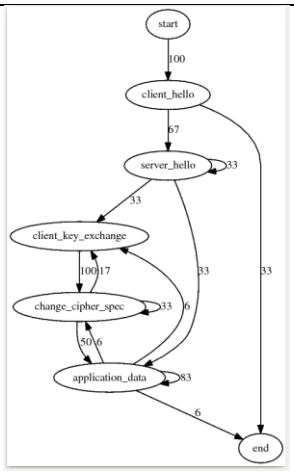
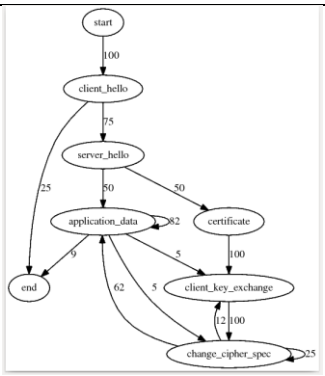
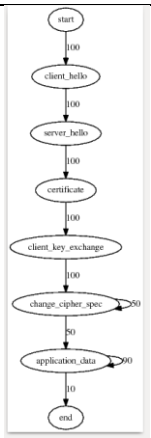
TeamViewer	GitHub Client	News App	Weather App	Skype
				

Table 1: Graph of TSL flows for 5 applications

The graphs represented are not perfect because scapy was not able to parse a noticeable quantity of packets, for example in the GitHub Client test the "certificate" element is completely missing.

Even considering this factor into count TeamViewer and Skype have a similar graph, the main difference is that in Skype the probability of staying in application_data is 90% and leave towards end 10% whereas TeamViewer has 50% of staying and leaving to alert.

Another similitude is given by the News and Weather application, (supposing that in News after server_hello there is a certificate) this is because both applications use an identical connection to the same Microsoft server to download the data.

Task 3

The total parsed handshake are 198 hence there were 5 blocks of 38 each;

The distance are the following : 1.029, 0.34, 0.46, 0.63, 1.47