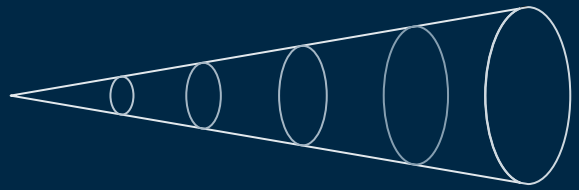
Network Anomaly Detection

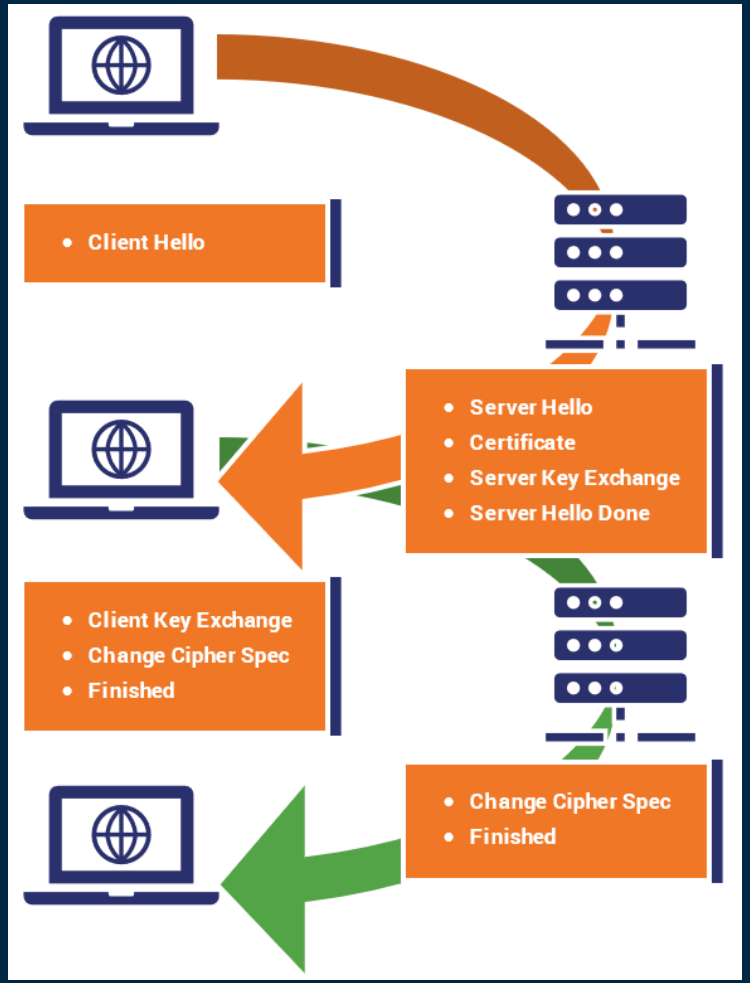
Jerry Lau

Introduction

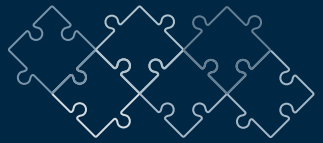
- Analyse network traffic flow
- Develop modules to monitor/fingerprint packets
- Use module to solve network related issue
- Check for anomalies or issues
- Deduce application methods by visualizing gathered data



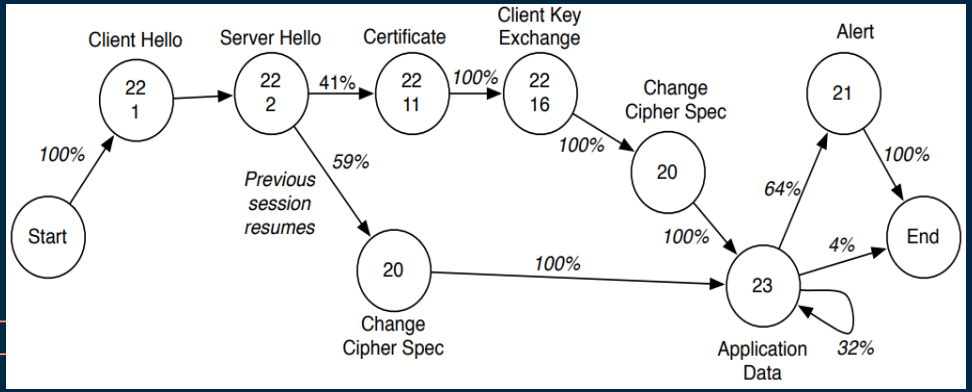
TLS negotiation



Project Objective



- Build a module to fingerprint TLS flow
- Aggregate data generated by module
- Launch TCP RST attack



Description & Methodology

- Record 1st byte in TLS handshake protocol
- Build dataset of various applications
- Visualize ways application implement TLS handshake
- Decide if some applications are better protected against RST attacks

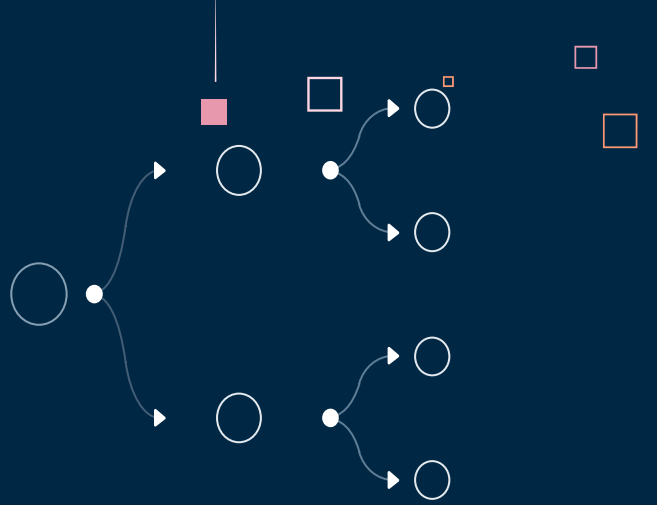
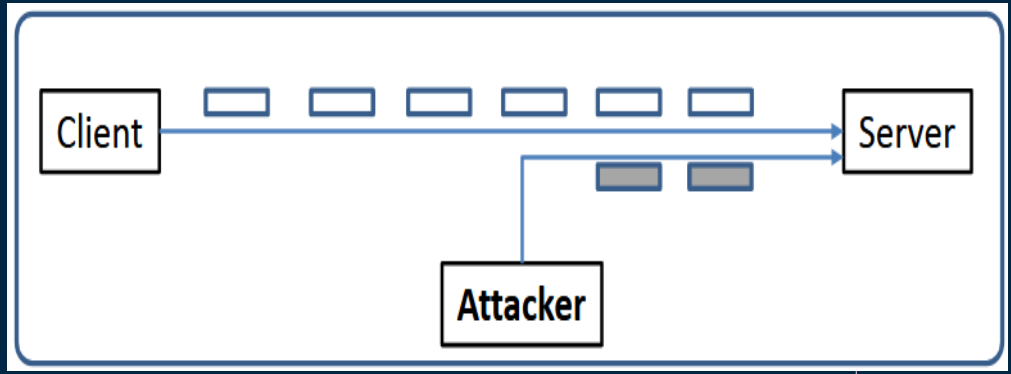


Table 1. The first byte in the SSL record payload belonging to the handshake protocol reveals which stage of the handshake is being performed through the record.

Handshake Message Type	Byte	Decimal
hello_request	0x00	0
client_hello	0x01	1
server_hello	0x02	2
certificate	0x0b	11
server_key_exchange	0x0c	12
certificate_request	0x0d	13
server_done	0x0e	14
certificate_verify	0x0f	15
client_key_exchange	0x10	16
finished	0x14	20

TCP Session Hijacking

- Attack user session via IP spoofing
- Possible due to the authentication process at the start of a TCP session



Resources

- 2 VM (1 client, 1 atk)
- Netwox
- Scapy ->pcap
- WireShark

Learning Experience

- Review and understand SSL/TLS protocol
- Build python programs
- Launch DOS attacks

Deliverables

- Implement monitoring module of TLS handshake
- Use module to visualize TLS process of various applications
- Launch RST attack

References/Bibliography

- Information based on a Dutch university advance network lab
  - Reduced size/scope
  - https://raw.githubusercontent.com/umeer/AdvancedNetworkSecurityProjects/master/Project%205/Project%20description/Project%205%20-%20Description.pdf
- SEED Lab TCP attacks
  - Followed the section about session hijacking

Thank You