



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Зворотна розробка та аналіз шкідливого програмного забезпечення

Лабораторна робота №5

Аналіз мережевих комунікацій

Мета:

Отримати навички аналізу мережевих комунікацій ШПЗ.

Перевірив:

Виконав:

студент III курсу

групи ФБ-01

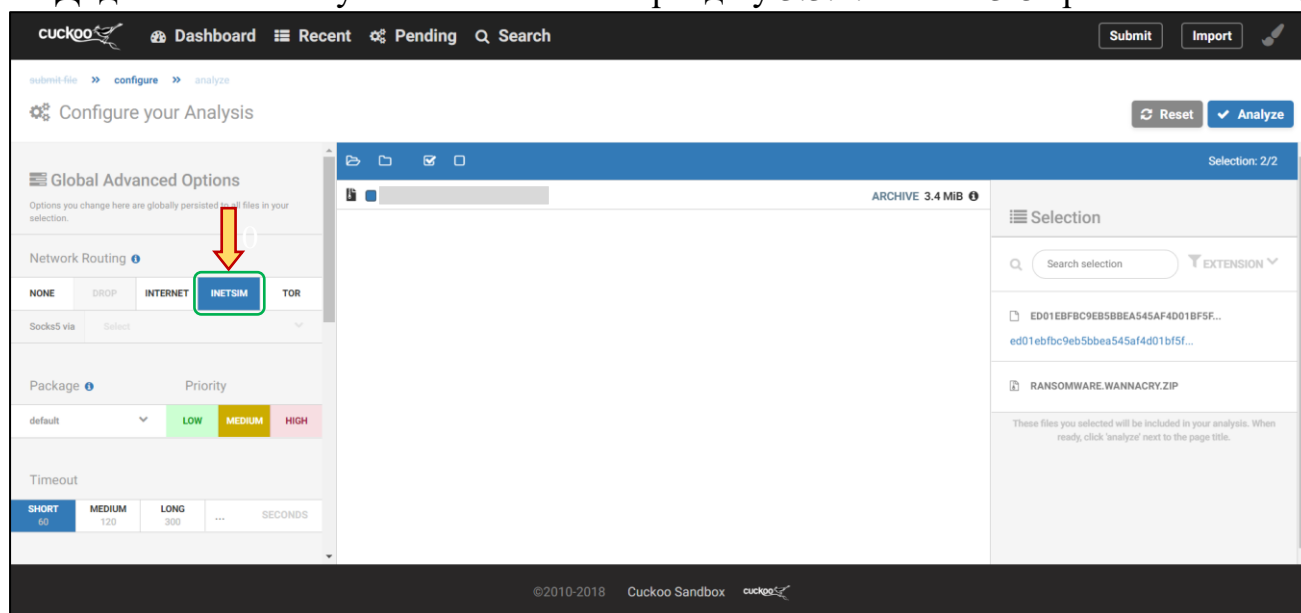
Сахній Н.Р.

Київ 2022

ФБ-01 Сахній Назар

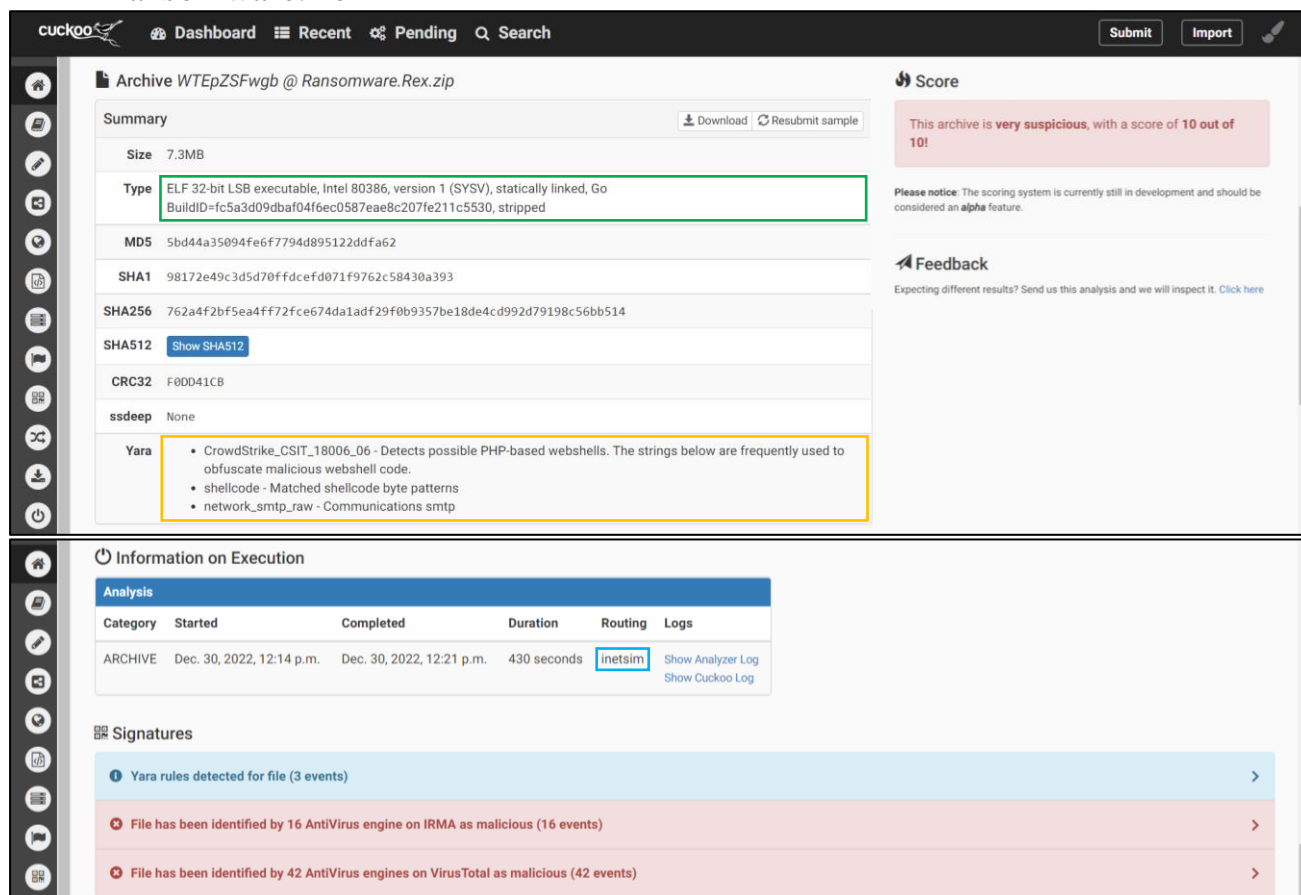
Завдання для виконання:

- Додання INetSim у Cuckoo Sandbox з розділу 3.3.1. Аналіз 3-5 зразків з theZoo.



↑ Обираємо емуляцію інтернет сервісів за допомогою INetSim у Cuckoo Sandbox

➤ Ransomware.Rex



➤ Proteus

Dashboard

Recent

Pending

Search

Submit

Import

Archive *gchrome.exe @ Proteus.zip*

Summary

Download

Resubmit sample

Size	2.8MB
Type	PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
MD5	49fd4020bf4d7bd23956ea892e6860e9
SHA1	c5d8f155209badd278437d0e534648f8d5c35aae
SHA256	d23b4a30f6b1f083ce86ef9d8ff434056865f6973f12cb075647d013906f51a2
SHA512	Show SHA512
CRC32	69D2AAC1
ssdeep	None
Yara	None matched

Score

This archive is **very suspicious**, with a score of **10 out of 10!**

Please notice: The scoring system is currently still in development and should be considered an **alpha** feature.

Feedback

Expecting different results? Send us this analysis and we will inspect it. [Click here](#)

Information on Execution

Analysis

Category	Started	Completed	Duration	Routing	Logs
ARCHIVE	Dec. 30, 2022, 12:13 p.m.	Dec. 30, 2022, 12:16 p.m.	208 seconds	inetsim	Show Analyzer Log Show Cuckoo Log

Signatures

The binary likely contains encrypted or compressed data indicative of a packer (2 events)

File has been identified by 15 AntiVirus engine on IRMA as malicious (15 events)

File has been identified by 56 AntiVirus engines on VirusTotal as malicious (50 out of 56 events)

➤ Win32.Triton

Dashboard

Recent

Pending

Search

Submit

Import

Archive *Win32.Triton @ Win32.Triton.zip*

Summary

Download

Resubmit sample

Size	84.0KB
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	1904cad4927541e47d453becbd934bf0
SHA1	aafa932eda97859e2b72772a3a8581760e860a46
SHA256	70efbd074326e7bbd4e851ded5c362fe5fe06282ed4bbb4b9f761f1b12ee32f7
SHA512	Show SHA512
CRC32	135E4C7D
ssdeep	None
Yara	<ul style="list-style-type: none">DebuggerCheck__QueryInfo - (no description)DebuggerHiding__Thread - (no description)anti_dbg - Checks if being debuggedwin_files_operation - Affect private profile

Score

This archive is **very suspicious**, with a score of **10 out of 10!**

Please notice: The scoring system is currently still in development and should be considered an **alpha** feature.

Feedback

Expecting different results? Send us this analysis and we will inspect it. [Click here](#)

Information on Execution

Analysis

Category	Started	Completed	Duration	Routing	Logs
ARCHIVE	Dec. 30, 2022, 12:13 p.m.	Dec. 30, 2022, 12:14 p.m.	48 seconds	inetsim	Show Analyzer Log Show Cuckoo Log

Signatures

Yara rules detected for file (4 events)

File has been identified by 14 AntiVirus engine on IRMA as malicious (14 events)

File has been identified by 59 AntiVirus engines on VirusTotal as malicious (50 out of 59 events)

➤ Ransomware.WannaCry

Archive ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe @ Ransomware.WannaCry.zip

Summary

- Size: 3.4MB
- Type: PE32 executable (GUI) Intel 80386, for MS Windows
- MDS: 84c82835a5d21bbcf75a61706d8ab549
- SHA1: 5ff465afaabcf0150d1a3ab2c2e74f3a4426467
- SHA256: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
- SHA512: [Show SHA512](#)
- CRC32: 4022FCAA
- ssdeep: None
- Yara:
 - WannaDecryptor - Detection for common strings of WannaDecryptor
 - Wanna_Sample_84c82835a5d21bbcf75a61706d8ab549 - Specific sample match for WannaCry
 - ransom_telefonica - Ransomware Telefonica
 - WannaCry_Ransomware_Generic - Detects WannaCry Ransomware on Disk and in Virtual Page
 - WannaCry_Ransomware - Detects WannaCry Ransomware
 - WannaCry_Ransomware_Dropper - WannaCry Ransomware Dropper
 - wannacry_static_ransom - Detects WannaCry spreaded during 2017-May-12th campaign and variants
 - WannaCry_Ransomware - Detects WannaCry Ransomware
 - CrowdStrike_CSIT_17102_03 - WannaCry ransomware, encrypted file header
 - Win32_Ransomware_WannaCry - Yara rule that detects WannaCry ransomware.

Score

This archive is **very suspicious**, with a score of 10 out of 10!

Feedback

Expecting different results? Send us this analysis and we will inspect it. [Click here](#)

Information on Execution

Category	Started	Completed	Duration	Routing	Logs
ARCHIVE	Dec. 30, 2022, 12:13 p.m.	Dec. 30, 2022, 12:17 p.m.	211 seconds	inetsim	Show Analyzer Log Show Cuckoo Log

Signatures

- Yara rules detected for file (10 events)
- The executable uses a known packer (1 event)
- The file contains an unknown PE resource name possibly indicative of a packer (1 event)
- The binary likely contains encrypted or compressed data indicative of a packer (2 events)
- File has been identified by 16 AntiVirus engine on IRMA as malicious (16 events)
- File has been identified by 64 AntiVirus engines on VirusTotal as malicious (50 out of 64 events)

- Розгортання OpenVPN за допомогою [openvpn-install](#), робота за протоколом TCP. Встановлення з'єднання на стороні клієнта з OpenVPN сервером через HTTP проксі. Проксі отримуємо за допомогою [fetch-some-proxies](#) або онлайн сервісів.

```
(nazar@snz24) - [/home/nazar/KPI/RevEng/Lab5_Report]
$ curl -O https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
% Total    % Received % Xferd  Average Speed   Time    Time     Current
           % Done    Dload  Upload    Total   Spent    Left   Speed
100 40586  100 40586    0     0  92141    0  --:--:-- --:--:-- --:--:--  92662

(nazar@snz24) - [/home/nazar/KPI/RevEng/Lab5_Report]
$ chmod +x openvpn-install.sh

(nazar@snz24) - [/home/nazar/KPI/RevEng/Lab5_Report]
$ sudo ./openvpn-install.sh
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

I need to ask you a few questions before starting the setup.
You can leave the default options and just press enter if you are ok with them.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address: 192.168.124.128

It seems this server is behind NAT. What is its public IPv4 address or hostname?
We need it for the clients to connect to the server.
Public IPv4 address or hostname: 54.76.30.11
```

↑ Спочатку отримуємо скрипт і зробимо його виконуваним, а далі вже запустимо

```
.....
What port do you want OpenVPN to listen to?
1) Default: 1194
2) Custom
3) Random [49152-65535]
Port choice [1-3]: 1

What protocol do you want OpenVPN to use?
UDP is faster. Unless it is not available, you shouldn't use TCP.
1) UDP
2) TCP
Protocol [1-2]: 2
```

```
.....
Write out database with 1 new entries
Data Base Updated

Client sakhnii added.

The configuration file has been written to /home/nazar/sakhnii.ovpn.
Download the .ovpn file and import it in your OpenVPN client.
```

Використаємо простий пайтон-скрипт `fetch.py` для отримання проксі-сервера ↓

```
(nazar@snz24) - [/home/nazar/KPI/RevEng/Lab5_Report]
$ ./fetch.py
+++++
f|e|t|c|h|_|s|o|m|e|_|p|r|o|x|i|e|s| <- v3.2.4
+++++

[i] initial testing...
[i] retrieving list of proxies...
[i] testing 1589 proxies (20 threads)...

socks4://178.48.68.61:4145 # latency: 0.86 sec; country: Hungary; anonymity: elite (high)
http://74.82.50.155:3128 # latency: 1.92 sec; country: United States; anonymity: transparent (low)
socks4://178.48.68.61:4145 # latency: 0.86 sec; country: Hungary; anonymity: elite (high)

[!] Ctrl-C pressed
```

Отже, встановимо з'єднання на стороні клієнта з OpenVPN сервером через HTTP проксі, використовуючи при цьому попередньо сконфігурований файл з'єднання.

```
(nazar@snz24) - [/home/nazar/KPI/RevEng/Lab5_Report]
$ sudo openvpn --config /home/nazar/sakhnii.ovpn --http-proxy 74.82.50.155 3128
2022-12-31 11:24:48 Unrecognized option or missing or extra parameter(s) in /home/nazar/sakhnii.ovpn:18: block-outside-dns (2.5.7)
2022-12-31 11:24:48 OpenVPN 2.5.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PTINF0] [AEAD] built on Jul  5 2022
2022-12-31 11:24:48 library versions: OpenSSL 3.0.4 21 Jun 2022, LZO 2.10
2022-12-31 11:24:48 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2022-12-31 11:24:48 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2022-12-31 11:24:48 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2022-12-31 11:24:48 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2022-12-31 11:24:48 TCP/UDP: Preserving recently used remote address: [AF_INET]74.82.50.155:3128
2022-12-31 11:24:48 Socket Buffers: R=[131072->131072] S=[16384->16384]
2022-12-31 11:24:48 Attempting to establish TCP connection with [AF_INET]74.82.50.155:3128 [nonblock]
2022-12-31 11:24:48 TCP connection established with [AF_INET]74.82.50.155:3128
2022-12-31 11:24:48 Send to HTTP proxy: 'CONNECT 178.158.203.247:1194 HTTP/1.0'
2022-12-31 11:24:48 Send to HTTP proxy: 'Host: 178.158.203.247'
2022-12-31 11:24:50 HTTP proxy returned: 'HTTP/1.1 200 Connection established'
2022-12-31 11:24:52 TCP_CLIENT link local: (not bound)
2022-12-31 11:24:52 TCP_CLIENT link remote: [AF_INET]74.82.50.155:3128
2022-12-31 11:24:52 TLS: Initial packet from [AF_INET]198.7.58.147:80, sid=9a95dcc7 5928ce34
2022-12-31 11:24:52 VERIFY OK: depth=1, CN=cn_RTxf412d6skTFsp
2022-12-31 11:24:52 VERIFY US OK
2022-12-31 11:24:52 Validating certificate extended key usage

.....
2022-12-31 11:24:56 /sbin/ip route add 128.0.0.0/1 via 10.12.0.17
2022-12-31 11:24:56 /sbin/ip route add 10.12.0.1/32 via 10.12.0.17
2022-12-31 11:24:56 Initialization Sequence Completed
```

Зробимо перевірку встановленого з'єднання за допомогою сайту whoer.net

WHOERMy IPVPNServersDownloadServices ~Grab -75% Now

My IP: 74.82.50.155

Secure internet

ISP:	US Dedicated	DNS	74.82.50.15 ³ United States
Hostname:	N/A	Proxy:	Yes
OS:	Linux	Anonymizer:	Yes
Browser:	Firefox 102.0	Blacklist:	No

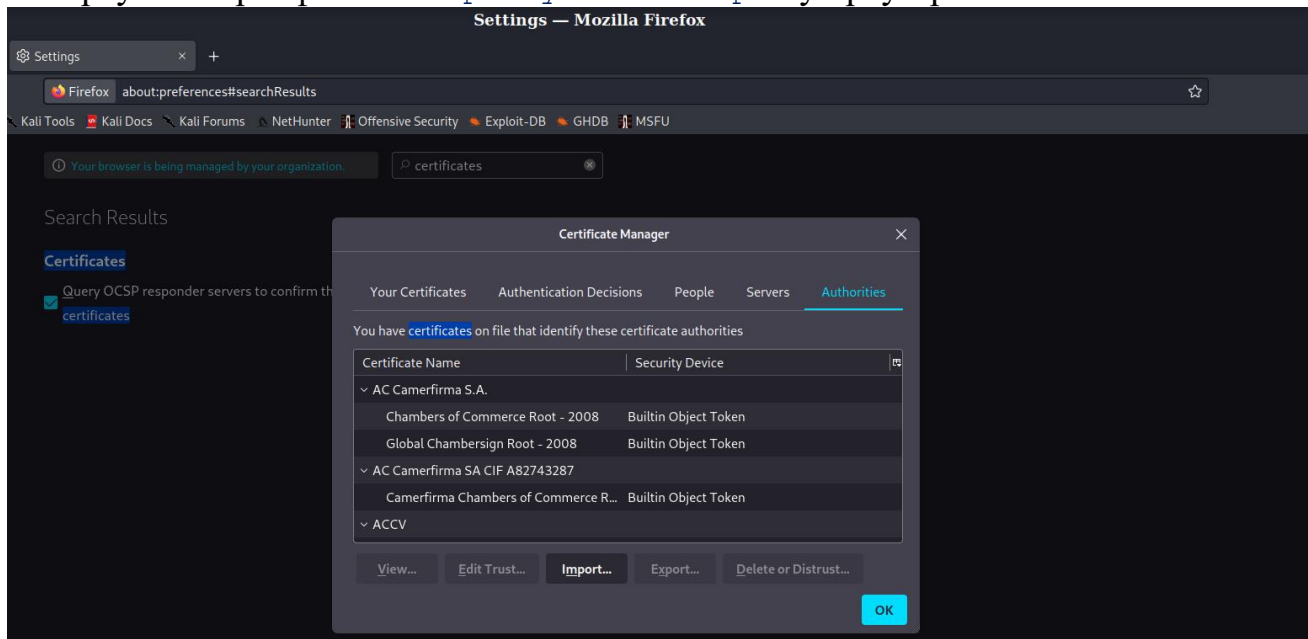
- Додання сертифікату СА mitmproxу у список довірених на клієнті. Аналіз трафіку власного зразку з лабораторної роботи 4.

```
nazar@snz24:~$ cd /home/nazar/KPI/RevEng/Lab5_Report
$ sudo apt-get install mitmproxy
```

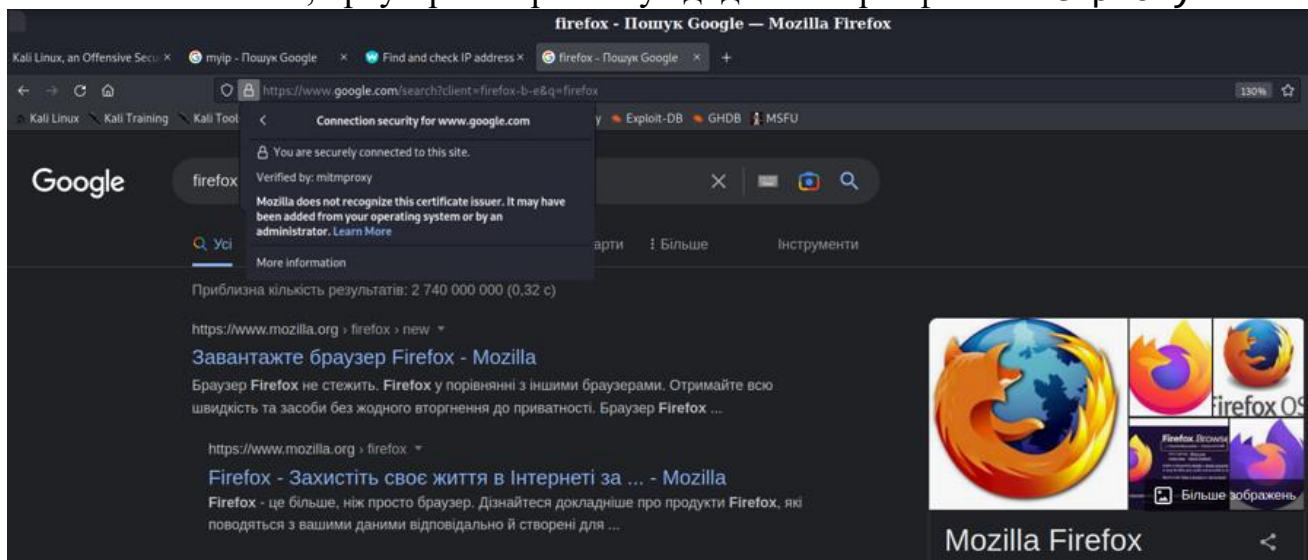
Для генерації сертифікату необхідно хоча б раз запустити mitmproxу:

```
nazar@snz24:~$ cd /home/nazar/KPI/RevEng/Lab5_Report
$ mitmproxy
```

Імпортуємо сертифікат `mitmproxy-ca-cert.pem` у браузер Firefox



Як можемо бачити, браузер використовує доданий сертифікат mitmproxу

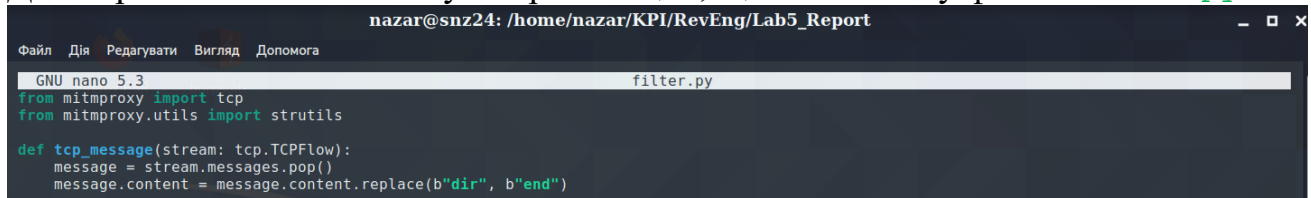


Для аналізу трафіка зразка із ЛР № 4 відповідно налаштуємо iptables, щоб трафік з портів 80 (HTTP) та 443 (HTTPS) йшов через mitmproxу (порт 8080).

```
nazar@snz24:~$ cd /home/nazar/KPI/RevEng/Lab5_Report
$ sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
nazar@snz24:~$ cd /home/nazar/KPI/RevEng/Lab5_Report
$ sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 8080
nazar@snz24:~$ cd /home/nazar/KPI/RevEng/Lab5_Report
$ mitmproxy --mode transparent
```

- Перенесення реалізації обробника пакетів на Python3 та запуск на шлюзі. Модифікація трафіку власного зразку з лабораторної роботи 4.

Для обробки пакетів застосуємо реалізацію, що міститься у файлі `filter.py`

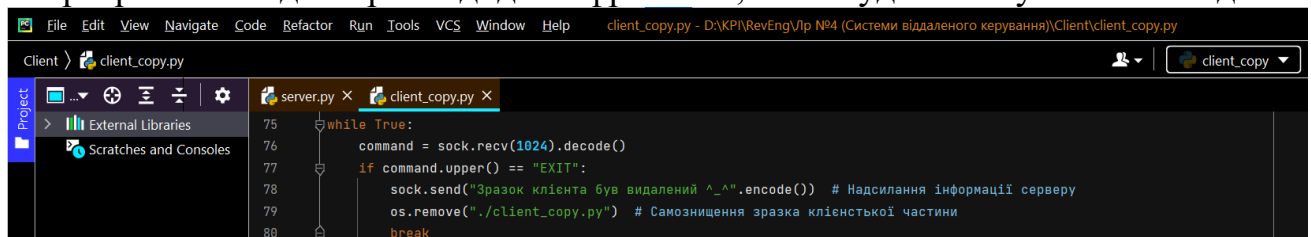


```
nazar@snz24: /home/nazar/KPI/RevEng/Lab5_Report
GNU nano 5.3 filter.py
from mitmproxy import tcp
from mitmproxy.utils import strutils

def tcp_message(stream: tcp.TCPFlow):
    message = stream.messages.pop()
    message.content = message.content.replace(b"dir", b"end")
```

- Розробка застосунку, що емулює (sinkhole) сервер керування для власного зразку з лабораторної роботи 4, – збирає інформацію про клієнта та подає команду самознищення (зразку, не цільової системи).

У програмний код із Лр №4 додамо фрагмент, який буде виконувати необхідне:



```
client_copy.py
75 while True:
76     command = sock.recv(1024).decode()
77     if command.upper() == "EXIT":
78         sock.send("Зразок клієнта був видалений ^_^".encode()) # Надсилання інформації серверу
79         os.remove("./client_copy.py") # Самознищення зразка клієнтської частини
80         break
```

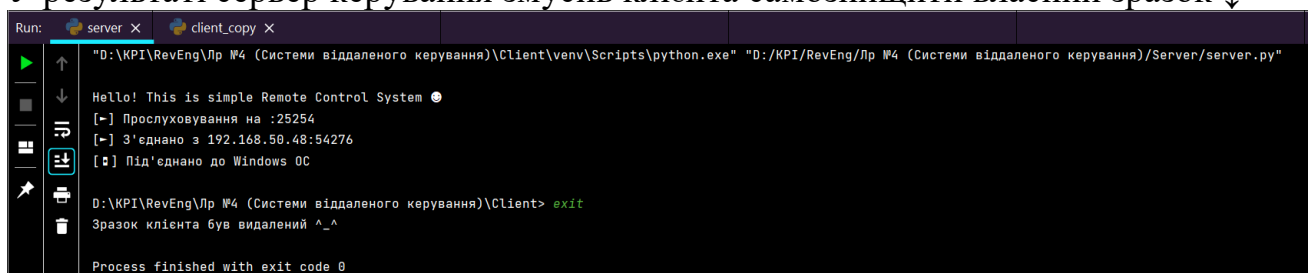
↑ Додано відповідний зразок програмного коду в копію клієнтської частини



```
server.py - server.py
82 if command == "exit":
83     info = client_socket.recv(1024).decode() # Сервер отримує повідомлення про знищення клієнтської частини
84     print(info)
85     break
```

↑ Відповідно й для реалізації серверної частини були внесені певні зміни

У результаті сервер керування змусив клієнта самознищити власний зразок ↓



```
Run: server X client_copy X
"D:\KPI\RevEng\Лр №4 (Системи віддаленого керування)\Client\venv\Scripts\python.exe" "D:\KPI\RevEng\Лр №4 (Системи віддаленого керування)\Server\server.py"
Hello! This is simple Remote Control System
[~] Прослуховування на :25254
[~] З'єднано з 192.168.50.48:54276
[*] Під'єднано до Windows OC
D:\KPI\RevEng\Лр №4 (Системи віддаленого керування)\Client> exit
Зразок клієнта був видалений ^_^
Process finished with exit code 0
```

Висновки:

У цій лабораторній роботі досліджувалися методи аналізу та протидії аналізу мережевого трафіку на прикладі зразків з ЛР №4 та відомого ШПЗ. Тому в результаті мною було отримано навички аналізу мережевих комунікацій ШПЗ та модифікації в реальному часі трафіку між зразком та центром керування.