

# Лабораторна робота №5

## Аналіз мережевих комунікацій

**Виконав:**

Студент 3 курсу ФТІ

групи ФІ-92

Поночевний Назар Юрійович

Варіант 6

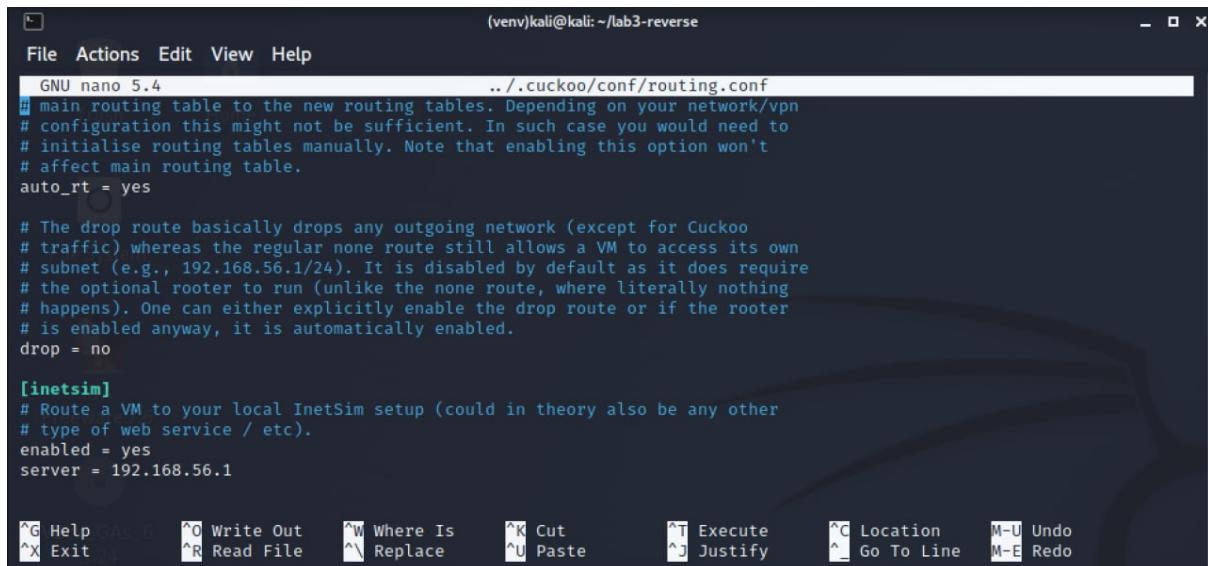
### Мета роботи

Отримати навички аналізу мережевих комунікацій ШПЗ.

### Завдання 1:

Додайте INetSim у Cuckoo Sandbox з розділу 3.3.1. Проаналізуйте 3-5 зразків з theZoo;

Спочатку під'єднати INetSim до Cuckoo:



```
(venv)kali㉿kali:~/lab3-reverse
File Actions Edit View Help
GNU nano 5.4 .. ./cuckoo/conf/routing.conf
# main routing table to the new routing tables. Depending on your network/vpn
# configuration this might not be sufficient. In such case you would need to
# initialise routing tables manually. Note that enabling this option won't
# affect main routing table.
auto_rt = yes

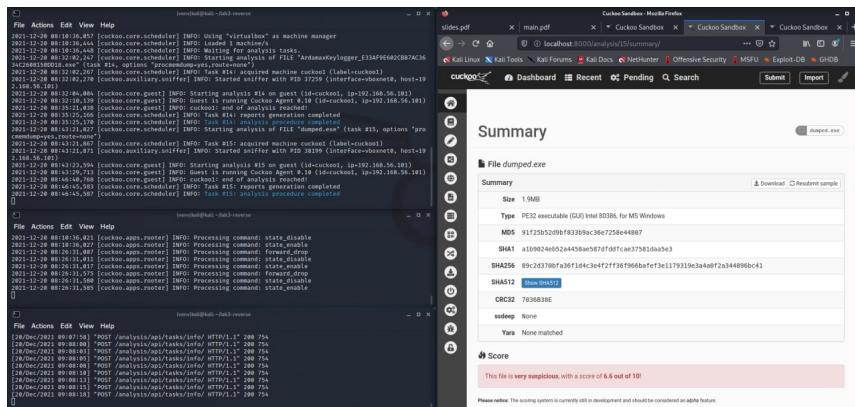
# The drop route basically drops any outgoing network (except for Cuckoo
# traffic) whereas the regular none route still allows a VM to access its own
# subnet (e.g., 192.168.56.1/24). It is disabled by default as it does require
# the optional rooter to run (unlike the none route, where literally nothing
# happens). One can either explicitly enable the drop route or if the rooter
# is enabled anyway, it is automatically enabled.
drop = no

[inetsim]
# Route a VM to your local InetSim setup (could in theory also be any other
# type of web service / etc).
enabled = yes
server = 192.168.56.1

^G Help ^A As^S 5 ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^D 24 ^R Read File ^Y Replace ^U Paste ^J Justify ^G Go To Line M-E Redo
```

Тепер проаналізуємо зразки:

#### 1) Kelihos



The screenshot shows the Cuckoo Sandbox interface with the following details:

- Summary:** The file is a PE32 executable (MS Windows) with MD5: 91C2505209B7833B9C36E25B644B07 and SHA1: A1092424052449385976710f7ca37581d0a5e3.
- Hashes:** SHA256: E9C2370fa1a8f1d4c3e4f2ff36f96bafe7e1179319c3a4a0f2a344896bc41, SHA512: 9b0540515052449385976710f7ca37581d0a5e3, CRC32: 78368383E.
- Score:** 6/10 (This file is very suspicious, with a score of 6.6 out of 10!).
- Logs:** The log window shows various system events and tasks from the analysis process.

The screenshot shows the Cuckoo Sandbox interface under the 'Network Analysis' tab. The top navigation bar includes 'Dashboard', 'Recent', 'Pending', 'Search', 'Submit', 'Import', and a gear icon. Below the navigation is a filter bar with tabs for Hosts (0), DNS (0), TCP (0), UDP (15), HTTP (0), ICMP (0), IRC (0), Suricata, and Snort. The 'UDP' tab is selected, showing 15 requests. A list of UDP requests is displayed:

- 192.168.56.100:67 → 192.168.56.101:68
- 192.168.56.101:137 → 192.168.56.255:137
- 192.168.56.101:138 → 192.168.56.255:138
- 192.168.56.101:49729 → 224.0.0.252:5355
- 192.168.56.101:51441 → 224.0.0.252:5355
- 192.168.56.101:52644 → 224.0.0.252:5355
- 192.168.56.101:52774 → 224.0.0.252:5355
- 192.168.56.101:52960 → 224.0.0.252:5355

To the right of the list is a detailed view of a specific UDP request from 192.168.56.100:67 to 192.168.56.101:68. It shows a hex dump of the packet payload:

```

192.168.56.100:67 → 192.168.56.101:68
[...]
plainext
hex
16 bytes 32 bytes 48 bytes
64 bytes
00000000: 0201 0600 eb4f ae17 0000 0000 0000 0000
00000010: 0000 0000 0000 0000 0000 0000 0000 2742
00000020: lca5 0000 0000 0000 0000 0000 0000 0000
00000030: 0000 0000 0000 0000 0000 0000 0000 0000
00000040: 0000 0000 0000 0000 0000 0000 0000 0000
00000050: 0000 0000 0000 0000 0000 0000 0000 0000
00000060: 0000 0000 0000 0000 0000 0000 0000 0000
00000070: 0000 0000 0000 0000 0000 0000 0000 0000
00000080: 0000 0000 0000 0000 0000 0000 0000 0000
00000090: 0000 0000 0000 0000 0000 0000 0000 0000
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000
000000e0: 0000 0000 0000 0000 0000 0000 0000 6382 5363
000000f0: 3604 c0a8 3864 3501 0501 04ff ffff 00ff

```

At the bottom left is a copyright notice: ©2010-2018 Cuckoo Sandbox. At the bottom right is a 'Back to Top' link.

## 2) Proteus

The screenshot shows the Cuckoo Sandbox interface with two open analysis windows. The top window is for 'slides.pdf' and the bottom window is for 'gchrome.exe'. Both windows show log files with numerous entries. The 'Summary' section for each file provides basic details like file type, size, and SHA256 hash.

**Summary for slides.pdf**

- Type: PDF executable (GUI) Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz
- Size: 2.8MB
- MD5: 49fd4802b14d7bd2395e6a92e686de9
- SHA1: c50bf1513209baed2784374be51464fb0d5c35aae
- SHA256: d2304a301601f003c056e79b0ff43485686576973f12cb0756470013900f51a2
- SHAS12: Show SHA12
- CRC32: 6902AAC1
- soddeep: None
- Yara: None matched

**Summary for gchrome.exe**

- Type: PE32 executable (GUI) Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz
- Size: 2.8MB
- MD5: 49fd4802b14d7bd2395e6a92e686de9
- SHA1: c50bf1513209baed2784374be51464fb0d5c35aae
- SHA256: d2304a301601f003c056e79b0ff43485686576973f12cb0756470013900f51a2
- SHAS12: Show SHA12
- CRC32: 6902AAC1
- soddeep: None
- Yara: None matched

A note at the bottom of the gchrome.exe summary states: "This file shows numerous signs of malicious behavior. The score of this file is 3.0 out of 10."

 Dashboard Recent Pending Search Submit Import



# Network Analysis

[Download pcap](#)

Hosts DNS TCP UDP 15 HTTP 1 ICMP 1 IRC 1 Suricata Snort

## UDP Requests

192.168.56.100:67	→	192.168.56.101:68
192.168.56.101:137	→	192.168.56.255:137
192.168.56.101:138	→	192.168.56.255:138
192.168.56.101:49729	→	224.0.0.252:5355
192.168.56.101:51441	→	224.0.0.252:5355
192.168.56.101:52644	→	224.0.0.252:5355
192.168.56.101:52774	→	224.0.0.252:5355
192.168.56.101:52960	→	224.0.0.252:5355

192.168.56.100:67 → 192.168.56.101:68

plaintext  
hex

16 bytes 32 bytes 48 bytes  
64 bytes

00000000: 0201 0600 cb4f ae17 0000 0000 0000 0000  
00000010: 0000 0000 0000 0000 0000 0000 0000 2742  
00000020: 1c4a 0000 0000 0000 0000 0000 0000 0000  
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  
000000e0: 0000 0000 0000 0000 0000 0000 6382 5363  
000000f0: 3604 c0a8 3864 3501 0501 04ff ffff 0fff

©2010-2018 Cuckoo Sandbox  Back to Top

### 3) Somoto

## Завдання 2:

Розгорніть OpenVPN за допомогою openvpn-install [102], робота за протоколом TCP. На стороні клієнта встановіть з'єднання з OpenVPN сервером через HTTP проксі. Прокси можна отримати за допомогою fetch-some-proxies [103] або онлайн сервісів;

```
Select an option [1-2]: 1

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/vars
Using SSL: openssl OpenSSL 1.1.1l 24 Aug 2021
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/easy-rsa-45100.GM9d1j/tmp.YBJJ5V'

Using configuration from /etc/openvpn/easy-rsa/pki/easy-rsa-45100.GM9d1j/tmp.U8q1Z1
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'kali-vpn'
Certificate is to be certified until Mar 24 15:50:28 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated
Client kali-vpn added.

The configuration file has been written to /home/kali/kali-vpn.ovpn.
Download the .ovpn file and import it in your OpenVPN client.

└─(kali㉿kali)-[~/lab4-reverse]
$ 1.24
```

```
(kali㉿kali)-[~/lab5-reverse/fetch-some-proxies]
$ sudo python fetch.py
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
||f||e||t||c||h||-||s||o||m||e||-||p||r||o||x||i||e||s|| ← v3.2.3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
[i] initial testing ...
[i] retrieving list of proxies...
[i] testing 2005 proxies (20 threads) ...

socks4://82.103.118.42:1099      # latency: 2.05 sec; country: Bulgaria; anonymity: elite (high)
socks4://188.163.170.130:35578   # latency: 1.21 sec; country: Ukraine; anonymity: elite (high)
http://152.136.39.231:80          # latency: 9.85 sec; country: China; anonymity: elite (high)
socks4://1.220.145.45:145        # latency: 1.71 sec; country: South Korea; anonymity: elite (high)
http://131.255.239.38:3128       # latency: 1.81 sec; country: Brazil; anonymity: elite (high)
socks4://45.65.18.4145           # latency: 1.02 sec; country: Spain; anonymity: elite (high)
http://218.104.169.78:80          # latency: 1.53 sec; country: China; anonymity: elite (high)
http://200.17.137.40:3128         # latency: 2.25 sec; country: Brazil; anonymity: elite (high)
http://169.57.157.148:8123       # latency: 1.53 sec; country: United States; anonymity: elite (high)
http://183.220.195.170:80         # latency: 2.97 sec; country: China; anonymity: elite (high)
http://169.57.157.148:8123       # latency: 1.25 sec; country: United States; anonymity: elite (high)
```

```
(root㉿kali)-[~/home/kali/lab5-reverse]
# openvpn --config /home/kali/kali-vpn.ovpn --http-proxy 89.109.7.67 443
1 ×
2021-12-20 11:27:46 Unrecognized option or missing or extra parameter(s) in /home/kali/kali-vpn.ovpn:18: block-outside-dns (2.5.1)
2021-12-20 11:27:46 OpenVPN 2.5.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on May 14 2021
2021-12-20 11:27:46 library versions: OpenSSL 1.1.1l 24 Aug 2021, LZO 2.10
2021-12-20 11:27:46 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2021-12-20 11:27:46 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2021-12-20 11:27:46 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2021-12-20 11:27:46 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2021-12-20 11:27:46 TCP/UDP: Preserving recently used remote address: [AF_INET]89.109.7.67:443
2021-12-20 11:27:46 Socket Buffers: R=[131072→131072] S=[16384→16384]
2021-12-20 11:27:46 Attempting to establish TCP connection with [AF_INET]89.109.7.67:443 [nonblock]
2021-12-20 11:27:46 TCP connection established with [AF_INET]89.109.7.67:443
2021-12-20 11:27:46 Send to HTTP proxy: 'CONNECT 5.181.248.224:1194 HTTP/1.0'
2021-12-20 11:27:46 Send to HTTP proxy: 'Host: 5.181.248.224'
```

### Завдання 3:

Додайте сертифікат CA mitmproxy у список довірених на клієнті (розділ 5.3.4).  
Проаналізуйте трафік Вашого зразку з лабораторної роботи 4;

The screenshot illustrates the steps to add a mitmproxy certificate authority to a browser. On the left, a terminal window shows the command to move the certificate file to the system's certificate store:

```
(kali㉿kali)-[~/tmp/mozilla_kali0]
$ sudo mv mitmproxy-ca-cert.pem /usr/local/share/ca-certificates/mitmproxy.crt
```

On the right, a Firefox browser window displays the "Install mitmproxy's Certificate Authority" page. It provides links for "Windows" and "Linux" to download the certificate file. The Linux section is shown in detail:

Install mitmproxy's Certificate Authority

**Windows**

**Linux**

Get mitmproxy-ca-cert.pem Show Instructions

After the certificate is added, the terminal shows the update process:

```
(kali㉿kali)-[~/tmp/mozilla_kali0]
$ sudo update-ca-certificates
Updating certificates in /etc/ssl/certs... done.
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d... done.
```

```
(kali㉿kali)-[~/Lab5-reverse]
$ sudo -u mitmproxyuser bash -c '/home/kali/lab5-reverse/mitmweb --mode transparent --showhost --set block_global=false -s /home/kali/Lab5-reverse/show.py'
Web server listening at http://127.0.0.1:8081/
No protocol specified
Unable to init server: Could not connect: Connection refused
Error: cannot open display: :0.0
Loading script /home/kali/lab5-reverse/show.py
Proxy server listening at http://*:8080
No protocol specified
Unable to init server: Could not connect: Connection refused
Error: cannot open display: :0.0
127.0.0.1:45726: client connect
127.0.0.1:45726: server connect 127.0.0.1:65432
[]

File Actions Edit View Help
kali@kali: ~/Lab5-reverse/rat

(kali㉿kali)-[~/Lab5-reverse/rat]
$ python3 server.py
Connected to (127.0.0.1, 65432)
USAGE: [command number] [args]
Supported command numbers:
"1" - system information discovery,
"2 [command] [args]" - command-line interface,
"3 [file/folder path]" - file and directory discovery,
"4 [your origin file path] [destination file for target]" - remote file copy,
"5 [file path]" - file deletion,
"6" - process discovery,
"7 [number of presses to capture]" - input capture,
"8" - clipboard data,
"9" - screen capture,
"10 [seconds to record]" - audio capture,
"11 [seconds to shot]" - video capture

> []
File Actions Edit View Help
kali@kali: ~/Lab5-reverse/rat

(kali㉿kali)-[~/Lab5-reverse/rat]
$ python3 client.py
POST /analysis/api/tasks/info/ HTTP/1.1" 200 758
```