ЗВОРОТНА РОЗРОБКА ТА АНАЛIЗ ШКIДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лабораторна робота №6

**Створiть парсер конфiгурацiї з пам’ятi Вашої системи з лабораторної роботи 4. Впевнiться, що парсер працює пiсля застосування UPX [124] та MPRESS [100] на виконуваному файлi зразку.**

import psutil

def print\_ip\_connections(process\_name):

for proc in psutil.process\_iter(['pid', 'name']):

if proc.info['name'] == process\_name:

pid = proc.info['pid']

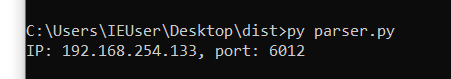
process = psutil.Process(pid)

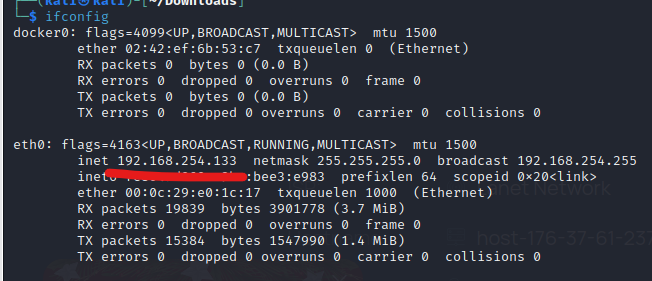
connections = process.connections()

for conn in connections:

print(f"IP: {conn.raddr.ip}, port: {conn.raddr.port}")

print\_ip\_connections('client.exe')





Я пробував зробити через winappdbg, але воно не видавало нормального результату

#!python2

import winappdbg

from winappdbg import win32, Process, HexDump

import re

s = winappdbg.System()

s.request\_debug\_privileges()

s.scan\_processes()

for p, path in s.find\_processes\_by\_filename("client.exe"):

pid = p.get\_pid()

bits = p.get\_bits()

print("pid %d (%d bits)" % (pid, bits))

mmap = p.get\_memory\_map()

mapf = p.get\_mapped\_filenames(mmap)

print(mmap)

print(mapf)

for m in mmap:

a = m.BaseAddress

fn = mapf.get(a, None)

if m.has\_content():

print("address 0x%x size 0x%x state 0x%x protect 0x%x type 0x%x[%s]" %

(a, m.RegionSize, m.State, m.Protect, m.Type, fn))

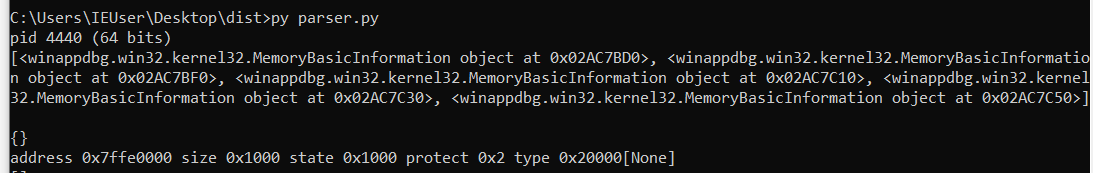
d = p.read(a, m.RegionSize)

cc = re.findall("\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}", d[::2])

print(cc)

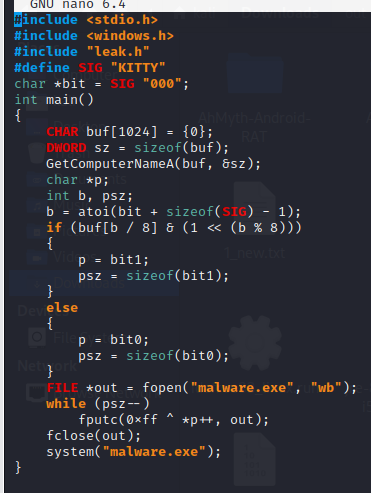
for c in cc:

print(c)

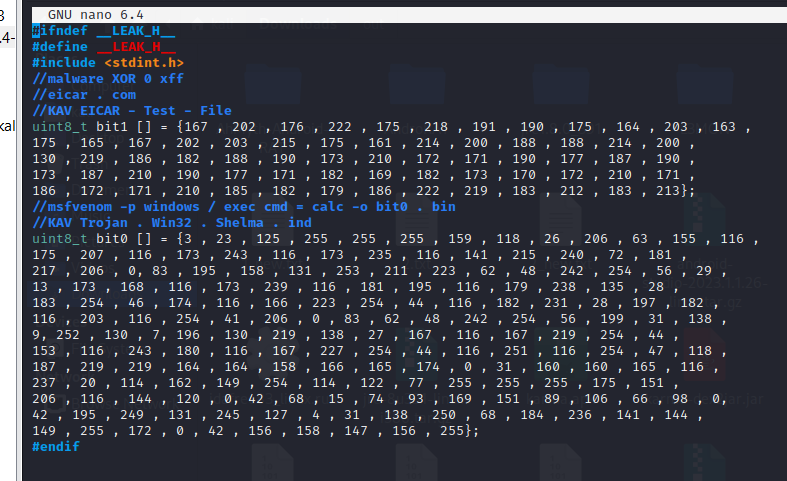


**• Проаналiзуйте 1-2 антивiруси з лабораторiї роздiлу 3.3.2 за допомогою методiв з роздiлу 6.3.3. Знайдiть iм’я системи, iм’я користувача, список процесiв, список файлiв на робочому столi, першi 32 байти notepad.exe. Для пришвидшення роботи рекомендується використати 256 зразкiв з theZoo [58], VirusShare [125] або iнших джерел [126] для отримання 1 байту за запит.**

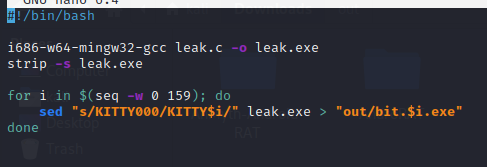
leak.c:



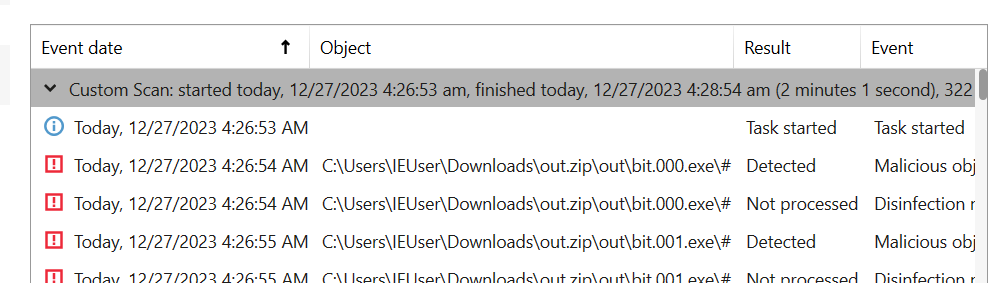
leak.h:

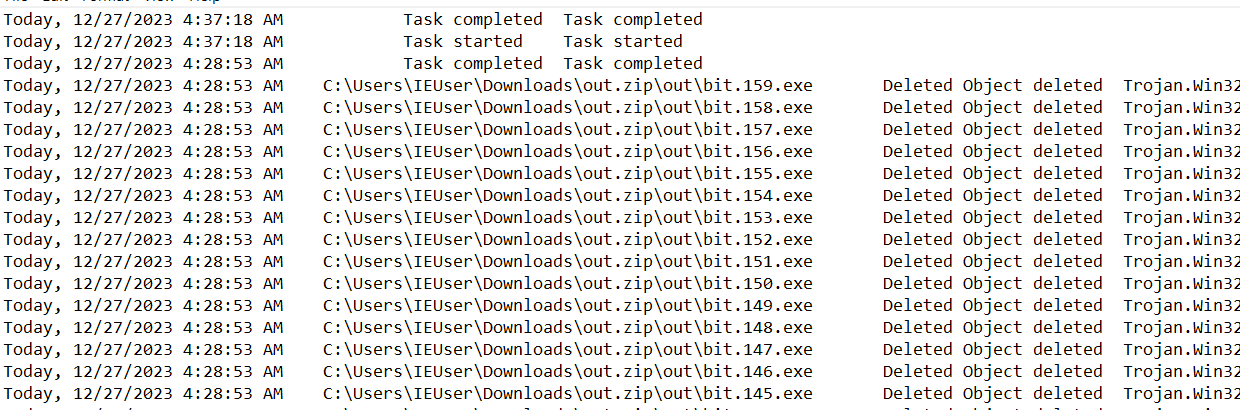
****

gen.sh



Створюємо 160 файлів, та скануємо архів за допомогою kaspersky

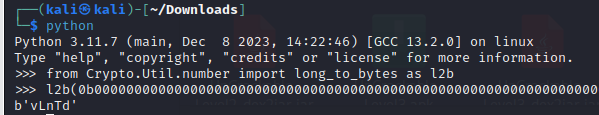




виділимо біти повідомлення:



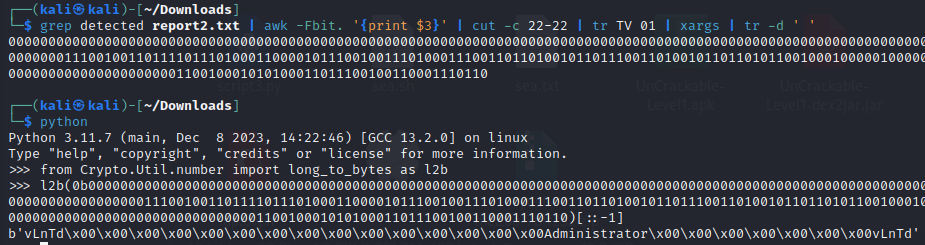
Отримуємо ім’я системи:



vLnTd

Створимо нові файли для отримання імені користувача(в leak.c GetComputerNameA() міняється на GetUserNameA())





Administrator