**1. Viết chương trình lây nhiễm virus vào tập tin thực thi PE (32-bit) trên Windows**

**a. Hiển thị thông điệp qua cửa sổ pop-up**

Viết một chương trình đơn giản để tạo cửa sổ pop-up với thông điệp "Infection by NT230" và nội dung thông điệp là "MSSV01\_MSSV02\_MSSV03".

from ctypes import windll

def popup\_message():

title = "Infection by NT230"

message = "MSSV01\_MSSV02\_MSSV03"

windll.user32.MessageBoxW(0, message, title, 1)

popup\_message()

**b. Hoàn trả chức năng gốc ban đầu của chương trình bị lây nhiễm**

Điều này liên quan đến việc lưu trữ chức năng gốc của chương trình và gọi lại nó sau khi thực thi payload. Ví dụ, nếu chúng ta có một chương trình gốc **original.exe**, chúng ta có thể sửa đổi chương trình virus để gọi lại chương trình gốc.

import os

def call\_original\_program():

os.system("original.exe")

def main():

popup\_message()

call\_original\_program()

main()

**2. Khả năng trốn tránh việc phát hiện**

**a. Nhận diện môi trường sandbox**

Cuckoo Sandbox là một môi trường phân tích malware phổ biến. Để nhận diện môi trường sandbox, ta có thể kiểm tra một số đặc điểm hệ thống mà chỉ xuất hiện trên sandbox.

import os

def is\_sandbox():

# Ví dụ đơn giản: kiểm tra tên máy

hostname = os.getenv('COMPUTERNAME')

sandbox\_names = ["mwsandbox", "cuckoosandbox"]

if hostname.lower() in sandbox\_names:

return True

return False

def main():

if not is\_sandbox():

popup\_message()

call\_original\_program()

else:

print("Running in sandbox environment. Exiting.")

main()

**b. Anti-debugging và anti-VM**

**Anti-Debugging:** Kiểm tra có debugger nào đang hoạt động.

import ctypes

def is\_debugged():

is\_debugger\_present = ctypes.windll.kernel32.IsDebuggerPresent()

return is\_debugger\_present != 0

def main():

if not is\_debugged():

popup\_message()

call\_original\_program()

else:

print("Debugger detected. Exiting.")

main()

**Anti-VM:** Kiểm tra nếu chương trình đang chạy trong máy ảo (VM).

import subprocess

def is\_vm():

output = subprocess.check\_output("systeminfo")

vm\_signs = ["VirtualBox", "VMware"]

for sign in vm\_signs:

if sign in output.decode():

return True

return False

def main():

if not is\_vm() and not is\_debugged() and not is\_sandbox():

popup\_message()

call\_original\_program()

else:

print("Analysis environment detected. Exiting.")

main()

**Câu a: Nhận diện môi trường sandbox**

**1. Tìm hiểu nguyên lý phát hiện sandbox**

**Sandbox là một môi trường ảo được sử dụng để phân tích hành vi của phần mềm. Mã độc có thể kiểm tra một số đặc điểm hệ thống để nhận diện nếu nó đang chạy trong môi trường sandbox, ví dụ:**

* **Tên hệ thống: Sandbox thường có tên hệ thống cụ thể như "Cuckoo", "Sandbox", v.v.**
* **File hệ thống và registry: Nhiều sandbox để lại các dấu hiệu đặc trưng trên hệ thống.**
* **Hoạt động hệ thống: Sandbox thường mô phỏng hành vi người dùng với một mức độ hạn chế.**

**2. Code nhận diện sandbox**

**Dưới đây là đoạn mã Python để kiểm tra nếu hệ thống đang chạy trong môi trường sandbox bằng cách kiểm tra tên hệ thống.**

import os

def is\_sandbox():

hostname = os.getenv('COMPUTERNAME')

sandbox\_names = ["mwsandbox", "cuckoosandbox"]

if hostname.lower() in sandbox\_names:

return True

return False

def popup\_message():

from ctypes import windll

title = "Infection by NT230"

message = "MSSV01\_MSSV02\_MSSV03"

windll.user32.MessageBoxW(0, message, title, 1)

def call\_original\_program():

os.system("original.exe")

def main():

if not is\_sandbox():

print("Not running in sandbox. Executing payload.")

popup\_message()

call\_original\_program()

else:

print("Running in sandbox. Exiting.")

main()

**Câu c: Tìm hiểu kỹ thuật Environmental Keying**

**1. Environmental Keying là gì?**

**Environmental Keying** là kỹ thuật trốn tránh phân tích dựa vào các đặc điểm môi trường thực tế. Mã độc sẽ chỉ hoạt động khi các điều kiện môi trường cụ thể được đáp ứng, như là một khóa bí mật.

**2. Phân tích các trường hợp điển hình**

**Equation Group** là một nhóm hacker được biết đến với các công cụ tấn công tiên tiến và phức tạp. Một trong các kỹ thuật mà họ sử dụng là Environmental Keying.

* **Cách hoạt động**: Equation sử dụng các khóa môi trường như thông tin hệ thống cụ thể, địa chỉ IP hoặc các phần cứng đặc thù để mã hóa và giải mã payload của họ.
* **Vai trò**: Điều này làm cho mã độc rất khó bị phân tích bởi vì nó chỉ hoạt động trong một môi trường rất cụ thể mà nó được thiết kế để nhắm đến.

**3. Sơ đồ lây nhiễm**

Dưới đây là sơ đồ lây nhiễm và giải thích ngắn gọn quy trình của kỹ thuật Environmental Keying:

+-------------------+

| Mã độc kích hoạt |

+-------------------+

|

v

+-------------------+

| Kiểm tra môi trường |

+-------------------+

|

v

+---------------------+

| Khóa môi trường đúng? |

+---------------------+

| |

Yes No

| |

v v

+---------------------+ +--------------------+

| Giải mã và thực thi | | Không thực thi và |

| payload | | thoát chương trình |

+---------------------+ +--------------------+

* **Mã độc kích hoạt**: Khi mã độc được kích hoạt, nó sẽ bắt đầu kiểm tra môi trường.
* **Kiểm tra môi trường**: Mã độc kiểm tra các điều kiện môi trường cụ thể