**BÀI TẬP 03***Môn học:* **Cơ chế hoạt động của mã độc***Tên chủ đề:* **Advanced Virus Techniques***Mã môn học: NT230 - Modus Operandi of Malware  
Học kỳ 2 - Năm học: 2023-2024*

*GVHD: Phan Thế Duy*

**Nhóm: G11**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT230.021.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email |
| 1 | Nguyễn Triệu Thiên Bảo | 21520155 | 21520155@gm.uit.edu.vn |
| 2 | Nguyễn Lê Thảo Ngọc | 21521191 | 21521191@gm.uit.edu.vn |
| 3 | Trần Lê Minh Ngọc | 21521195 | 21521195@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Tình trạng** | **Trang** |
| 1 | Yêu cầu a | 100% | 2 - 5 |
| 2 | Yêu cầu b |  |  |
| 3 | Yêu cầu c | 100% | 5-7 |

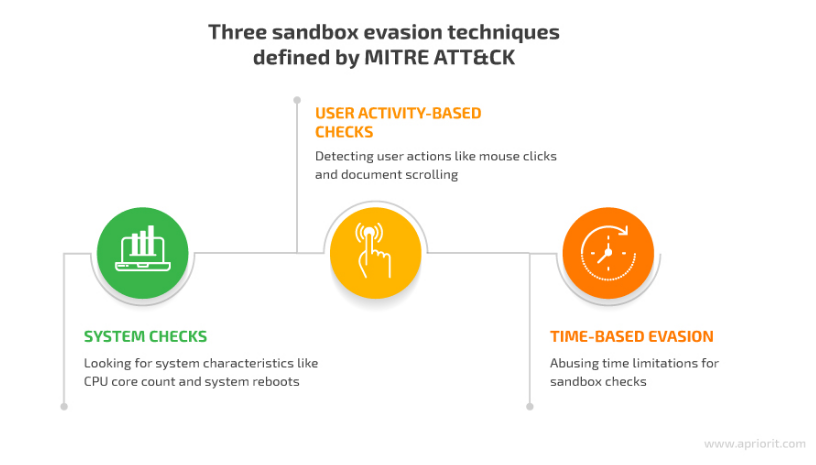
**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

**BÁO CÁO CHI TIẾT**

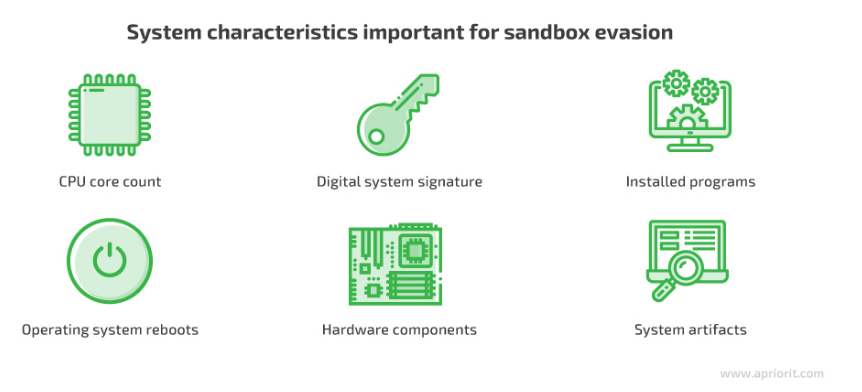
***Yêu cầu thực hiện***Viết chương trình lây nhiễm virus vào tập tin thực thi (tập tin thực thi trên Windows  
– PE file 32 bits) có tính năng đơn giản (mục đích demo giáo dục) như yêu cầu bên dưới.  
*Về chức năng, mục đích của payload (sử dụng lại phần virus cơ bản của bài tập 01):*- Hiển thị thông điệp ra màn hình thông qua cửa sổ “pop-up” với tiêu đề cửa sổ là  
“**Infection by NT230**” và cấu trúc thông điệp là “**MSSV01\_MSSV02\_MSSV03**”  
(thông tin MSSV của các thành viên trong nhóm). Lưu ý: không có dấu “”.  
- Hoàn trả chức năng gốc ban đầu của chương trình bị lây nhiễm (không phá hủy  
chức năng của chương trình vật chủ)

**a) Tìm hiểu nguyên lý phát hiện sandbox** (thí dụ như Cuckoo Sandbox,…). Trình  
bày ngắn gọn các kỹ thuật nhận diện sandbox để trốn tránh cho mã độc.

MITRE ATT&CK xác định các kỹ thuật nhận diện sandbox để trốn tránh cho mã độc phổ biến nhất là 3 loại như sau:



**1. System check (phát hiện thông qua tương tác hệ thống)**



Kỹ thuật phát hiện các tương tác hệ thống hoạt động dựa trên một số tính năng của hệ thống có trong môi trường máy thực nhưng không có trong sandbox hoặc môi trường ảo, có thể kể đến như:

- СPU core count and hardware component: Mã độc có thể tìm kiếm sự khác biệt giữa hệ thống ảo và thực, chẳng hạn như số lượng CPU core và lợi dụng việc này để phát hiện sandbox. Đây là lý do nhiều nhà cung cấp sandbox che thông tin cấu hình thực tế để tin tặc không thể phát hiện thông số kỹ thuật của sandbox.

- Digital system signature: Một số mã độc được thiết kế để tìm chữ ký số của hệ thống. Chữ ký số này có chứa thông tin về cấu hình của hệ thống.

- Installed programs : Kỹ thuật này cho phép mã độc kiểm tra tính khả dụng của các chương trình Antivirus bằng cách tìm kiếm các tiến trình hoạt động trong hệ điều hành. Mã độc Client Maximus là một ví dụ cho việc áp dụng kỹ thuật này để lẩn tránh sandbox bằng một trình driver ẩn mình.

- Operating system reboots : Trong một số sandbox, hệ thống không thể hoạt động khi khởi động lại. Dựa trên dấu hiệu này, một số mã độc được thiết kế chỉ kích hoạt khi hệ thống khởi động lại. Mặc dù môi trường ảo cố gắng mô phỏng việc khởi động lại bằng cách đăng nhập và đăng xuất người dùng, nhưng mã độc có thể phát hiện ra điều này vì không phải tất cả các kích hoạt khởi động lại đều được thực thi.

- Một máy ảo chứa các thành phần cụ thể trong bộ nhớ, tiến trình và file của chính nó. Mã độc có thể tìm các thành phần như chỉ thị máy ảo, sự giao tiếp với port I/O để phát hiên nó có ở trong sandbox không.

**2. User activity-based (phát hiện dựa trên tương tác người dùng)**



Trong quá trình sử dụng, người dùng tương tác với máy tính bằng nhiều thao tác khác nhau như nhấn bàn phím, cuộn hay nhấp chuột. Tuy nhiên, những tương tác này không xảy ra trong môi trường sandbox. Dựa trên đặc điểm này, tin tặc có thể thiết kế mã độc chờ đợi một hành động cụ thể từ người dùng và chỉ thực hiện hành vi độc hại sau khi phát hiện ra hành động đó. Ví dụ như:

- Cuộn chuột trong tài liệu: Một số mã độc hiện đại có thể được lập trình là chỉ thực thi sau khi cuộn đến một vị trí cụ thể trong tài liệu. Chẳng hạn, một mã độc chỉ được kích hoạt sau khi người dùng cuộn văn bản RTF sang trang thứ hai. Để phát hiện hành động này, mã độc phải chứa các mã đoạn văn bản được sử dụng trong các văn bản Microsoft Word. Trong khi các tệp RTF cũng bao gồm các dấu đoạn văn bản (./par), thì mã độc chứa một loạt các đoạn cần được cuộn trước khi thực thi mã khai thác. Môi trường sandbox không có bất kỳ chuyển động cuộn nào, do vậy mã độc sẽ không được kích hoạt và qua mặt được giải pháp bảo mật này.

- Di chuyển và nhấp chuột: Một cách khác, mã độc được lập trình để kiểm tra tốc độ di chuyển và nhấp chuột. Nếu tốc độ di chuột đáng ngờ, thì mã độc sẽ không được kích hoạt, như Trojan.APT.BaneChan chỉ kích hoạt sau một số lượng nhấp chuột nhất định của người dùng

-Lịch sử trình duyệt, cache, bookmark: Mỗi khi lướt internet, người dùng sẽ để lại lịch sử và cache trên máy tính của họ. Nhưng sanbox sẽ không lưu lại chúng bởi vì môi trường sanbox thường xuyên dọn dẹp cache và lịch sử trình duyệt để xoá đi các malware tiềm ẩn, cũng như đảm bảo sandbox hoạt động chính xác. Malware sẽ được lập trình để kiểm tra máy tính có cá thông tin này không. Nếu không tìm thấy các thông tin này thì nó sẽ bất động.

-Lưu file vào các thư mục cụ thể

Malware có thể kiểm tra hệ thống có cho phép lưu các file đã chỉnh sửa vào root folder và desktop không. Vì việc lưu file theo cách này sẽ không an toàn nên sandbox sẽ không cho phép việc này diễn ra nhưng ở môi trường khác như Windows thì việc này không bị cấm. Do đó nếu không thấy sự cho phép lưu các file đã chỉnh sửa vào root folder và desktop thì malware sẽ nằm im, không làm gì cả.

**3. Time-based evasion (phát hiện dựa trên thời gian)**



Trong một số trường hợp, mã độc lẩn tránh sandbox bằng cách sử dụng kỹ thuật dựa trên thời gian. Kỹ thuật này lợi dụng tính năng phân tích mã độc chỉ trong một khoảng thời gian giới hạn của sandbox.

Dưới đây là ba loại kỹ thuật lẩn tránh sandbox dựa trên thời gian phổ biến:

- Ngủ kéo dài: Khi mã độc sử dụng các lệnh ngủ kéo dài, nó có thể vượt qua sandbox thành công trước khi thực thi.

- Logic Bom: Trong một số trường hợp, mã độc có thể được lập trình để thực thi vào một ngày hay một thời điểm cụ thể.

- Mã trì hoãn: mã độc có thể chứa các đoạn mã thực hiện các chu kỳ CPU vô dụng để trì hoãn việc thực thi các đoạn code độc hại cho đến khi sandbox kết thúc quá trình kiểm tra.

**c) Nâng cao: Tìm hiểu kỹ thuật Environmental Keying – một trong những cơ chế trốn tránh phát hiện của mã độc:**+ *Nhóm sinh viên tìm hiểu cách thức hoạt động của kỹ thuật này thông qua phân tích các trường hợp báo cáo điển hình được thống kê trên MITRE. Ví dụ: Tìm hiểu cách thức hoạt động của một trong các loại mã độc sau: Equation, InvisiMole, Ninja, PowerPunch, APT41,…*+ *Trình bày báo cáo yêu cầu này dưới dạng sơ đồ lây nhiễm (kèm giải thích ngắn ngọn qui trình), nhấn mạnh vào vai trò của kỹ thuật Environmental Keying.*

Environmental Keying là một kỹ thuật bảo mật được sử dụng bởi kẻ tấn công để ràng buộc việc thực thi hoặc hành động của mã độc vào các điều kiện cụ thể của môi trường mục tiêu. Kỹ thuật này sử dụng các giá trị môi trường đặc thù của mục tiêu để tạo ra các khóa mã hóa/giải mã, giúp mã độc tránh bị phát hiện và phân tích ngược.

Để hiểu rõ hơn về cách thức hoạt động của Evironmental Keying, nhóm sẽ tiến hành phân tích mã độc APT41.

* Khái niệm:

APT41 là một nhóm đe dọa mạng mà các nhà nghiên cứu đánh giá là một nhóm gián điệp do nhà nước Trung Quốc tài trợ. APT41 sử dụng một kho vũ khí gồm hơn 46 loại mã độc và công cụ khác nhau để thực hiện các hoạt động tài chính. Nhóm gián điệp này đã hoạt động ít nhất từ năm 2012 và đã nhắm mục tiêu vào các ngành công nghiệp chăm sóc sức khỏe, viễn thông, công nghệ, và trò chơi điện tử ở 14 quốc gia.

* Cách thức hoạt động của APT41:

Initial Compromise

(Phising, supply chain compromise)

Command and Control

(Evironmental Keying)

Reconnaissance

(Thu thập thông tin mục tiêu)

Establish Foothold

(Thực hiện duy trì kết nối)

Actions on Objectives

(Cài đặt mã độc)

Giải thích:

* Reconnaissance (Trinh sát): APT41 thực hiện thu thập thông tin về mục tiêu, bao gồm email, hệ thống và mạng, sử dụng các công cụ để quét và xác định các điểm yếu của hệ thống mục tiêu.
* Initial Compromise (Xâm nhập ban đầu): APT41 sử dụng email lừa đảo (phising) hoặc tấn công chuỗi cung ứng (supply chain compromise) để xâm nhập vào hệ thống mục tiêu. Khi người dùng mục tiêu mở tệp đính kèm, mã độc được thực thi, cho phép APT41 xâm nhập vào hệ thống.
* Establish Foothold (Thiết lập chỗ đứng): Sau khi xâm nhập, APT41 triển khai các công cụ và mã độc để duy trì sự hiện diện trên hệ thống mục tiêu.
* Command and Control (Chỉ huy và kiểm soát)\*\*:
  + Environmental Keying: APT41 mã hóa các payload bằng cách sử dụng Data Protection API (DPAPI), dựa trên các khóa liên kết với các tài khoản người dùng cụ thể trên các máy cụ thể. Điều này đảm bảo rằng mã độc chỉ hoạt động trên môi trường mục tiêu cụ thể.
  + Second Stage Malware: APT41 cũng sử dụng kỹ thuật mã hóa môi trường cho mã độc giai đoạn hai với khóa RC5 được tạo một phần từ số seri của ổ đĩa hệ thống bị nhiễm. Kỹ thuật này giúp tránh việc phát hiện và phân tích mã độc.
* Actions on Objectives (Hành động theo mục tiêu): APT41 thực hiện các hoạt động nhằm đạt được mục tiêu cuối cùng của mình, bao gồm thu thập dữ liệu, phá hoại hệ thống, hoặc cài đặt thêm mã độc để duy trì sự hiện diện lâu dài. Các hành động này có thể bao gồm triển khai rootkits và bootkits MBR để tăng cường ẩn giấu và duy trì sự kiểm soát trên hệ thống.
* Vai trò của Environmental Keying:

Environmental Keying giúp mã hóa payload và mã độc của APT41 dựa trên các giá trị môi trường cụ thể của hệ thống mục tiêu, đảm bảo rằng mã độc chỉ hoạt động trên các hệ thống đã được định trước, từ đó tránh bị phát hiện và phân tích bởi các hệ thống bảo mật hoặc môi trường không phải mục tiêu.

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)