# BTL-Phân Tích Mã Độc-Lớp 01-Nhóm 07

Quản Lý An Toàn Thông Tin (Học viện Công nghệ Bưu chính Viễn thông)

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

# Môn: Phân tích mã độc

# Bài tập lớn đề tài: Phân tích tĩn


**Giảng viên giảng dạy**: Thầy Đỗ Xuân Chợ

**Lớp 01 – Nhóm 07:**

Bùi Minh Hoàng - B19DCAT078

Phạm Quang Huy - B20DCAT083

Bùi Đăng Phúc – B20DCAT139

Thân Văn Tiến - B20DCAT159


Hà Nội, tháng 11 năm 2023

# Mục lục

# I. CƠ SỞ LÝ THUYẾT

## 1. Tổng quan về mã độc

### 1. Mã độc là gì?

Mã độc (Malware hay Malicious software) là một loại phần mềm được tạo ra và chèn vào hệ thống một cách bí mật với mục đích thâm nhập, phá hoại hệ thống, lấy cắp thông tin, làm gián đoạn hoặc tổn hại tới tính bí mật, tính toàn vẹn và tính sẵn sàng của hệ thống hay các máy tính cá nhân.

Nó có thể có dạng một tập tin thực thi, script, mã nguồn hoặc bất kỳ phần mềm nào khác.

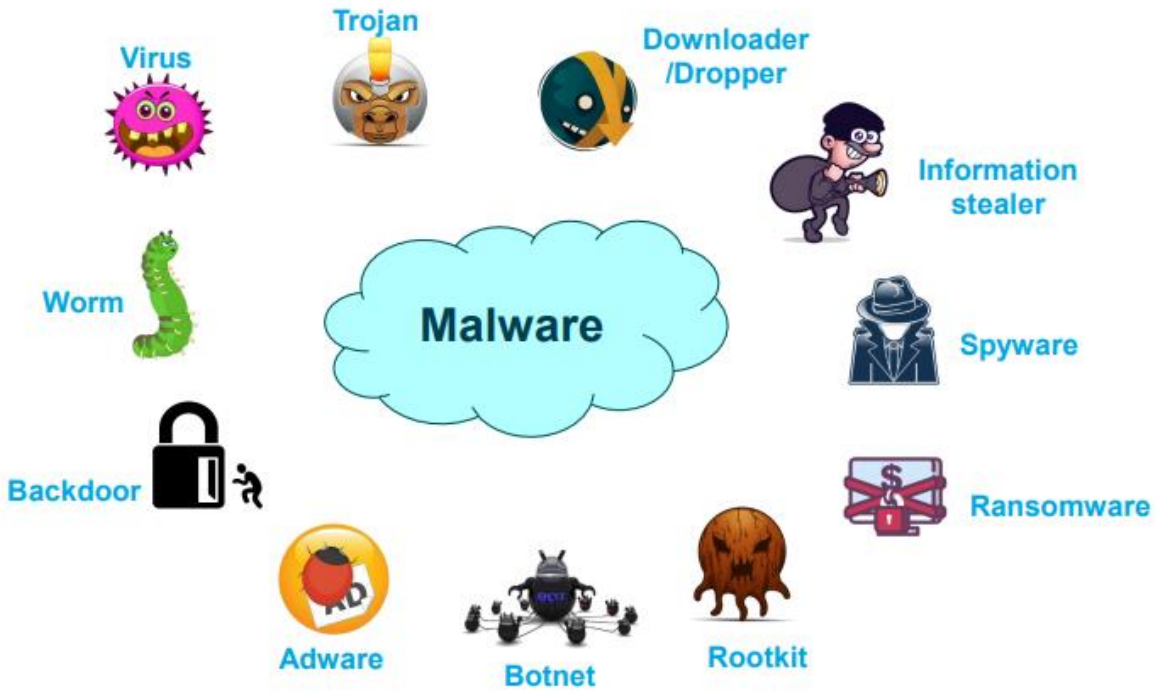
Kẻ tấn công sử dụng mã độc để đánh cắp thông tin nhạy cảm, giám sát hệ thống bị nhiễm và chiếm quyền kiểm soát hệ thống.

Thông thường, mã độc xâm nhập trái phép vào hệ thống của nạn nhân và có thể lây lan qua các kênh truyền thông khác nhau như email, web hoặc ổ đĩa USB.

Các mục tiêu chính nhất có thể được phân loại thành:

- ➢ Gây gián đoạn hoạt động của hệ thống máy chủ
- ➢ Đánh cắp thông tin quan trọng, chẳng hạn như thông tin cá nhân và tài chính
- ➢ Truy cập trái phép vào hệ thống hoặc tài khoản - Gián điệp - Gửi thư rác
- ➢ Sử dụng hệ thống của nạn nhân để thực hiện tấn công DDoS
- ➢ Khóa tệp tin của nạn nhân trên máy chủ và yêu cầu tiền chuộc để mở khóa.

### 2. Phân loại mã độc



**Bom logic** (Logic bombs): mã độc này thường được "nhúng" vào các chương trình bình thường và thường hẹn giờ để "phát nổ" trong một số điều kiện cụ thể. Khi "phát nổ" bom logic có thể xoá dữ liệu, files, tắt cả hệ thống...

**Trojan** Horses: Chứa mã độc, thường giả danh những chương trình có ích, nhằm lừa người dùng kích hoạt chúng.

**Back door** (cửa hậu): Thường được các lập trình viên tạo ra, dùng để gỡ rối và test chương trình. Cửa hậu thường cho phép truy nhập trực tiếp vào hệ thống mà không qua các thủ tục kiểm tra an ninh thông thường. Do đó nó trở thành một mối đe dọa đến an ninh hệ thống.

**Rootkit:** Là một loại phần mềm độc hại được thiết kế để ẩn danh trên hệ thống máy tính và cho phép kẻ tấn công có quyền truy cập cao nhất vào hệ thống mà không bị phát hiện. Rootkit thường được sử dụng để giấu các hoạt động độc hại khác trên hệ thống, như truy cập trái phép, sao chép dữ liệu hoặc thực hiện các cuộc tấn công khác.

**Adware** (tên đầy đủ là advertising supported software): Là các phần mềm tự động hiển thị các bảng quảng cáo trong thời gian người dùng tải hoặc sử dụng các phần mềm. Adware thường được đóng gói chung với các phần mềm khác có thể dưới dạng như một phần của một phần mềm hoặc một dịch vụ miễn phí.

**Spyware:** Là một dạng phần mềm độc hại được cài đặt tự động nhằm giám sát, thu thập và đánh cắp các thông tin nhạy cảm trên hệ thống nạn nhân.

**Ransomware:** Là một loại phần mềm độc hại có khả năng mã hoá các tệp tin trên hệ thống của nạn nhân và yêu cầu nạn nhân phải trả tiền chuộc để nhận được chìa khóa giải mã. Ransomware là một trong những mối đe dọa an ninh mạng nguy hiểm nhất hiện nay và có thể gây thiệt hại nặng nề cho các tổ chức và cá nhân.

**Vi rút** (Virus): Là một chương trình có thể "nhiễm" vào các chương trình khác, bằng cách sửa đổi các chương trình này. Nếu các chương trình đã bị sửa đổi chứa vi rút được kích hoạt thì virus sẽ tiếp tục "lây nhiễm" sang các chương trình khác. Vi rút máy tính cũng có khả năng tự nhân bản, tự lây nhiễm sang các chương trình khác mà nó tiếp xúc. Có nhiều con đường lây nhiễm vi rút, như sao chép file, gọi các ứng dụng và dịch vụ qua mạng, email...

**Sâu** (Worm): Là một loại phần mềm độc hại có khả năng tự lây nhiễm từ máy này sang máy khác mà không cần chương trình chủ, vật chủ, hoặc sự trợ giúp của người dùng. Khi sâu lây nhiễm vào một máy, nó sử dụng máy này làm "bàn đạp" để tiếp tục rà quét, tấn công các máy khác. Các phương pháp lây lan chính của sâu gồm: lây lan qua thư điện tử, lây lan thông qua khả năng thực thi từ xa, lây lan thông qua khả năng log-in (đăng nhập) từ xa.

**Zombie/Bot:** Là một chương trình được thiết kế để giành quyền kiểm soát một máy tính, hoặc thiết bị tính toán có kết nối Internet và sử dụng máy tính bị kiểm soát để tấn công các hệ thống khác, hoặc gửi thư rác. Botnet (mạng máy tính ma) là một tập hợp các máy tính bot dưới sự kiểm soát của một, hoặc một nhóm kẻ tấn công.

### 3. Nguyên tắc hoạt động của mã độc

a. 5 nguyên tắc hoạt động chung của mã độc bao gồm:

Xâm nhập vào hệ thống: Mã độc thường sử dụng các lỗ hổng bảo mật hoặc phương pháp xâm nhập khác để xâm nhập vào hệ thống máy tính hoặc thiết bị khác.

Điều khiển và kiểm soát hệ thống: Sau khi xâm nhập thành công, mã độc sẽ tìm cách lấy quyền điều khiển và kiểm soát hệ thống. Điều này cho phép kẻ tấn công có thể thực hiện các hoạt động độc hại trên hệ thống mà không bị phát hiện.

Mở cửa hậu cho kẻ tấn công: Mã độc thường sẽ tạo ra các lỗ hổng hoặc cửa sau trên hệ thống cho phép kẻ tấn công có thể tiếp tục tấn công hoặc truy cập vào hệ thống sau này mà không cần phải xâm nhập lại.

Thực hiện các hoạt động độc hại: Mã độc thường sẽ thực hiện các hoạt động độc hại trên hệ thống, bao gồm việc tạo ra các file giả mạo, thay đổi các cài đặt hệ thống, mã hoá các tệp tin quan trọng, đánh cắp thông tin cá nhân hoặc thông tin quan trọng, và thậm chí là điều khiển các thiết bị khác trong mạng của nạn nhân.

Che dấu hoạt động: Mã độc thường sẽ giấu kín hoạt động của mình để tránh bị phát hiện và gỡ bỏ khỏi hệ thống. Điều này có thể bao gồm việc tránh các chương trình chống virus, mã hóa các tệp tin hoặc các kết nối mạng bằng các phương thức mã hóa chéo, hoặc thay đổi các cài đặt hệ thống để tránh bị phát hiện.

b. Hình thức phát tán:

- Phát tán qua email
- Phát tán qua usb
- Phát tán qua lỗ hổng
- Phát tán qua điểm yếu hệ thống
- Phát tán qua các file nguồn không an toàn: crack, keygen, free, social

### 4. Các định dạng dữ liệu nhiễm mã độc

| Stt | Vật chủ | Loại virus | Các định dạng | Kiểu |
|---|---|---|---|---|
| 1 | Tập tin văn bản | File virus<br>Worm<br>Trojan | Tập tin lô | BAT |
| | | | Tập tin script | VBS, JS |
| | | | Tập tin registry | REG |
| | | | Tập tin siêu văn bản | HTT, HTA |
| 2 | Tập tin chương trình | File virus<br>Worm<br>Trojan | Tập tin lệnh | COM |
| | | | Tập tin thi hành | EXE, SCR |
| | | | Tập tin thư viện | DLL, CPL, SYS, VXD |
| 3 | Tập tin MS Office | Macro virus | Tập tin tư liệu | DOC, DOT |
| | | | Tập tin bảng tính | XLS, XLT |
| | | | Tập tin trình diễn | PPT, POT |
| 4 | Mẫu tin khởi động | Boot virus | Mẫu tin khởi động hệ điều hành đĩa mềm | #N/A |
| | | | Mẫu tin khởi động hệ điều hành đĩa cứng | #N/A |
| | | | Mẫu tin khởi tạo phân khu đĩa cứng | #N/A |

## 5. Các hành vi & kỹ thuật ẩn mình và dấu hiệu cơ bản của mã độc

### *a. Thành phần của mã độc*

Window:

- File
- Registry
- Keys
- Processes memory
- Folder

Linux:

- File
- Processes memory
- Folder

***Chi tiết:***

Files: Đây là thành phần chính của mã độc, chứa mã nguồn, encrypt data, payload và các thông tin của mã độc…

Registry keys là các khóa trong hệ thống registry trên Windows, chứa các thông tin quan trọng về hệ thống và các ứng dụng. Mã độc có thể tạo, ghi, sửa đổi các key để khởi động và thực thi cùng hệ thống.

Process, memory: Mã độc có thể tạo ra các process (tiến trình) mới hoặc sửa đổi bộ nhớ (memory) của các tiến trình đang chạy trên máy tính để thực hiện các hoạt động độc hại.

Foders: Một số mã độc tạo ra các thư mục, thư mục ẩn để che dấu các tệp tin độc hại.

***Kỹ thuật ẩn mình***

Che giấu: Sử dụng các kỹ thuật mã hóa, ẩn file.

- File có thuộc tính ẩn (hidden file, system file): Theo mặc định, windows thiết lập không hiển thị các tệp có thuộc tính ẩn hoặc thuộc tính tệp hệ thống (system files) Mã độc thường lợi dụng cơ chế này để ẩn các file độc hại.
- Giả mạo icon: Mã độc có thể giả mạo icon của thư mục file nén phần mềm chuẩn.
- Giả mạo shortcut.
- Giả mạo file.

Rootkit: Mã độc ẩn đi các thành phần của chính nó.

- Fileless: Mã độc Fileless không tồn tại dạng file trong hệ thống.
- Chèn mã, system call: Mã độc thực thi qua các lời gọi hệ thống hoặc chèn mã vào các tiến trình có sẵn.
- Mã độc có thể lợi dụng các tiến trình, service hệ thống để ẩn mình.
- Khai báo lỗ hổng:Mã độc sử dụng các kỹ thuật và lỗ hổng mới
- Virus lây file: mã độc ẩn mình bằng cách lây nhiễm vào các chương trình thực thi.

***Dấu hiệu nhận biết:***

- Máy tính chạy chậm bất thường, chậm kết nối mạng
- Máy tính bị khóa hoặc không trả lời (stop responding) liên tục, không cho chạy các chương trình hệ thống (cmd, regedit, task manager, gpedit, run,..)
- Máy tính tự động khởi động lại hoặc bị lỗi (crashes).
- Khi chạy một chương trình thường thông báo lỗi, chạy các file *.exe, *.com,... đều bị thay thế bởi các chương trình khác.
- Ẩn file, thư mục, tạo các thư mục lạ, các biểu tượng lạ. Xuất hiện icon mới hoặc icon cũ tự mất.
- Xuất hiện các cửa sổ pop-up hoặc thông báo lạ, những tin nhắn báo lỗi không bình thường.
- Hiển thị hoặc file in ra bị biến dạng.
- Xuất hiện cặp đôi phần mở rộng của file. Ví dụ: vbs.txt...
- Phần mềm diệt virus không chạy hoặc không thể cài đặt.
- Tệp bị lỗi hoặc thư mục được tạo ra một cách tự động hoặc bị thay đổi, bị xóa, bị mã hóa.
- Hệ thống bị thay đổi cài đặt hay bị kiểm soát từ xa

## 6. Các công cụ rà quét mã độc và các phương pháp phòng chống mã độc

Công cụ rà quét

- File: Explorer, cmd…

- Process: Task manager, Process Explorer, Process Hacker…
- Network: TcpView, Wireshark…
- Startup: Regedit, AutoRuns…
- Rootkit: PC Hunter, Rootkit Remover…
- Logs: Event Viewer, Process Monitor…
- Scanner: KVRT, TDSSKiller, Norton Power Eraser, ClamAV…

Phương pháp phòng chống

## 2. Tổng quan về phân tích mã độc

### 2.1. Phân tích mã độc là gì?

Phân tích mã độc là việc nghiên cứu hành vi của mã độc.

Mục tiêu của phân tích mã độc là hiểu cách thức hoạt động của mã độc và cách phát hiện và loại bỏ nó.

### 2.2. Vai trò của phân tích mã độc

Mục đích chính khi thực hiện phân tích mã độc là thu thập thông tin từ mẫu mã độc. Từ đó xác định được khả năng của mã độc, phát hiện và ngăn chặn nó.

Ngoài ra phân tích mã độc cũng làm nguồn cung cấp, xác định mẫu mã độc để hỗ trợ việc phát hiện và ngăn chặn trong tương lai.

Các nguyên nhân chính để thực hiện phân tích mã độc như sau:

- Xác định bản chất và mục đích của mã độc. Ví dụ, xác định liệu mã độc có phải là kẻ đánh cắp thông tin, bot HTTP, bot spam, rootkit, keylogger, RAT...
- Hiểu rõ hệ thống bị xâm phạm như thế nào và ảnh hưởng của nó.

Xác định các chỉ số mạng liên quan đến mã độc, sau đó có thể sử dụng để phát hiện mã độc tương tự bằng cách theo dõi mạng.

Thu thập các chỉ số dựa trên máy chủ như tên tập tin, khóa registry, sau đó có thể sử dụng để xác định mã độc tương tự bằng cách theo dõi trên máy chủ.

Xác định ý định và động cơ của kẻ tấn công. Ví dụ, trong quá trình phân tích, nếu thấy mã độc đánh cắp thông tin ngân hàng  động cơ của kẻ tấn công là tiền.

### 2.3. Phân loại kỹ thuật phân tích mã độc

#### ❖ Phân tích tĩnh

Phân tích mã nguồn hoặc file thực thi của mã độc mà không cho mã độc hoạt động. Thường thực hiện trên máy tính an toàn không kết nối mạng. Cho phép khám phá cấu trúc và chức năng bên trong của mã độc. Phát hiện các công cụ và kỹ thuật được sử dụng.

Ưu điểm: An toàn hơn, dễ thực hiện.

Nhược điểm: Khó nắm bắt hành vi thực tế của mã độc.

Phân loại:

- Phân tích tĩnh cơ bản: tìm cách hiểu mã độc bằng cách phân tích chính file, cấu trúc file, các chức năng được sử dụng bởi mã độc ...
- Phân tích tĩnh nâng cao: phân tích sâu hơn và tìm cách hiểu được mã độc dựa trên dịch ngược (disassembled).

❖ **Phân tích động**

Phân tích mã độc bằng cách cho mã độc thực thi và giám sát và phân tích hoạt động của nó. Thường thực hiện trong môi trường cô lập để đảm bảo an toàn. Cho phép nghiên cứu hành vi của mã độc trên máy chủ, tương tác với mạng và hệ thống. Có thể phát hiện các chỉ số mạng và máy chủ liên quan đến mã độc.

Ưu điểm: Hiểu được hành vi và tác động của mã độc.

Nhược điểm: Cần môi trường cô lập và có thể nguy hiểm.

Phân loại:

- Phân tích động cơ bản: Chạy mã độc trong môi trường cô lập có trang bị sẵn các công cụ giám sát khác nhau và cố gắng hiểu mã độc đang làm gì dựa trên đầu ra của các công cụ đó.
- Phân tích động nâng cao: nếu phân tích cơ bản không đem lại kết quả hoặc muốn tìm hiểu kỹ hơn thì cần thực hiện phân tích nâng cao mã đ bằng cách sử dụng một bộ gỡ lỗi (debugger). Bằng cách này, chuyên gia có nhiều quyền kiểm soát hơn về cách mã độc được thực thi.

3. **Phân tích tĩnh**

### 3.1. Phân tích tĩnh là gì?

Phân tích tĩnh là kỹ thuật phân tích tệp nghi ngờ (mẫu) mà không thực thi nó.

Là phương pháp phân tích ban đầu liên quan đến việc trích xuất thông tin hữu ích từ tệp nhị phân nghi ngờ để đưa ra quyết định có căn cứ về cách phân loại hoặc phân tích mã độc và chỉ ra hướng phân tích tiếp theo.

Phân tích tĩnh được chia làm 2 loại:

a) Phân tích tĩnh cơ bản.Phân tích tĩnh cơ bản bao gồm các kỹ thuật như:

Giải mã: Giải mã mã nhị phân thành mã nguồn lập trình.

- Phân tích chuỗi: Tìm kiếm các chuỗi, hằng số trong mã.
- Phân tích cấu trúc: Phân tích cấu trúc gói, lớp, hàm trong mã.

b) Phân tích tĩnh nâng cao. Phân tích tĩnh nâng cao bao gồm các kỹ thuật phức tạp hơn như:

- Phân tích luồng điều khiển: Tìm hiểu luồng điều khiển của chương trình.
- Phân tích dòng dữ liệu: Theo dõi dòng dữ liệu qua chương trình.
- Phân tích hàm: Phân tích các hàm trong chương trình để hiểu chức năng.

### 3.2. Vai trò của phân tích tĩnh

Cung cấp thông tin ban đầu mà không cần thực thi tệp: Phân tích tĩnh có thể được thực hiện mà không cần thực thi tệp. Điều này cho phép thu thập thông tin ban đầu về tệp mà không kích hoạt mã độc tiềm ẩn.

Xác định hành vi tiềm ẩn: Phân tích tĩnh có thể tiết lộ những gợi ý về hành vi tiềm ẩn của tệp như kết nối mạng, truy cập hệ thống tập tin, v.v. Điều này có thể giúp xác định liệu tệp có đáng nghi ngờ hay không.

Xác định cách thức hoạt động: Phân tích tĩnh có thể tiết lộ cách thức hoạt động của tệp thông qua việc khám phá cấu trúc, luồng điều khiển và các hàm.

Hỗ trợ phân tích động sau đó: Thông tin thu thập được từ phân tích tĩnh có thể hỗ trợ phân tích động sau đó khi thực thi tệp trong môi trường cô lập.

Phân loại tệp: Phân tích tĩnh có thể cung cấp đủ thông tin để phân loại nhanh tệp là bening hoặc đáng ngờ.

### 3.3. Một số công cụ phân tích tĩnh

***IDA***

***GHIDRA***

***NICore***

***SSDEEP***

***Capa***: Capa tập trung vào việc xác định các đặc điểm quan trọng của mã độc, phát hiện chữ ký. Ngoài ra, đầu ra của chúng cũng dễ đọc cho những nhà phân tích

***Floss***: Cũng giống như Capa, công cụ Floss cũng tìm chữ ký trong tệp thực thi, ngoài ra nó còn tìm chuỗi. Nhưng nếu với một số con mã độc có chuỗi ký tự ngoài chuỗi ký tự Floss nhận biết được, thì nó không thể tìm thấy do bị giới hạn trong việc xác định các loại mã độc ngoài chuỗi ký tự.

***Cutter***: Công cụ có khả năng thực hiện debugging cho các tệp thực thi

**PEStudio**: Công cụ này cung cấp thông tin chi tiết về các thành phần quan trọng trong tệp thực thi: module, ký sự số bản quyền, thông tin nhúng giúp kiểm tra tệp PE. Ngoài nó, nó có khả năng phát hiện các dấu hiệu của mã độc trong tệp thực thi. Đây là một công cụ phân tích tĩnh hiệu quả, nhưng nó giới hạn trong phân tích động do không theo dõi được các hoạt động của tệp thực thi.

***CFF Explorer***: Công cụ này cung cấp thông tin một cách chi tiết về các cấu trúc của tệp thực thi, bao gồm module, bảng phân đoạn ,.. Ngoài ra nó còn cho phép xem và chỉnh

sửa các phần quan trọng trong PE headers. Đây là một công cụ tốt hỗ trợ trong việc phân tích tĩnh, đặc biệt trong phần xác định tệp PE, nhưng đối với các nhà phân tích muốn tích hợp cả việc phân tích tĩnh và động, thì nó không cung cấp khả năng phân tích động

### 3.4. Quy trình phân tích tĩnh

#### 3.4.1. Xác định loại của tệp

Phần mở rộng tệp (File extension) là một phần của tên tệp được gắn liền với tên tệp để chỉ định định dạng hoặc loại tệp, vd: .exe, .dll, .sys, docx, xlsx... Phần mở rộng tệp giúp hệ điều hành và các ứng dụng phần mềm nhận biết loại tệp và sử dụng chương trình mở tương ứng để xem hoặc xử lý tệp. Tuy nhiên, các kẻ tấn công có thể sử dụng các chiêu trò khác nhau để che dấu tệp của họ bằng cách sửa đổi phần mở rộng tệp và thay đổi giao diện của tệp để lừa người dùng vào thực thi nó.

Chữ ký tệp (File Signature) có thể được sử dụng để xác định loại tệp, thay cho phần mở rộng tệp. Chữ ký tệp là trình tự byte duy nhất được viết vào phần đầu tệp. Các tệp khác nhau có chữ ký khác nhau có thể được sử dụng để xác định loại tệp. VD: Các tệp thực thi Windows, còn gọi là tệp PE (như các tệp kết thúc bằng .exe, .dll, .com, .drv, .sys ...) có chữ ký tệp là MZ hoặc ký tự thập lục phân 4D 5A trong hai byte đầu tiên của tệp.

Timestamp: Phần mềm đóng gói chứa thông tin xác định thời điểm tệp được được biên dịch; kiểm tra trường này có thể cho bạn biết khi phần mềm độc hại được tạo lần đầu. Thông tin này có thể hữu ích trong việc xây dựng một bản thời gian của chiến dịch tấn công. Cũng có khả năng rằng một kẻ tấn công đã sửa đổi dấu thời gian để ngăn một nhà phân tích biết thời điểm thực sự. Dấu thời gian biên dịch đôi khi có thể được sử dụng để phân loại các mẫu đáng ngờ.

Các công cụ sử dụng để xác định loại của tệp:

- Sử dụng phương pháp thủ công để xác định loại tệp là tìm kiếm chữ ký tệp bằng cách mở nó trong một trình chỉnh sửa hex. Trình chỉnh hex là một công cụ cho phép một nhà điều tra kiểm tra từng byte của tệp; hầu hết trình chỉnh hex cung cấp nhiều chức năng giúp trong việc phân tích tệp.
- Sử dụng phương pháp xác định bằng công cụ
    - ➢ Trên hệ điều hành Linux có thể sử dụng công cụ file utility.

- ➢ Trên hệ điều hành Windows có thể sử dụng công cụ CFF Explorer, là 1 phần của công cụ Explorer Suite có thể được sử dụng để xác định loại tệp; nó không chỉ giới hạn trong việc xác định loại tệp. Đây cũng là một công cụ tuyệt vời để kiểm tra các tệp thực thi (cả 32-bit và 64-bit) và cho phép bạn xem xét cấu trúc nội bộ của tệp PE, sửa đổi các trường và trích xuất tài nguyên.
- ➢ Sử dụng phương pháp xác định bằng ngôn ngữ Python. Sử dụng thư viện python-magic

### 3.4.2. Phân tích mã Hash

Quá trình tạo các giá trị băm cho các tệp nghi ngờ dựa trên nội dung của chúng, cũng giống như tạo vân tay (Fingerprinting) cho mã độc. Các thuật toán băm thường được sử dụng như MD5, SHA1 hoặc SHA256. Sử dụng các giá trị băm cho phân tích mã độc mang lại các lợi thế sau:

Định danh duy nhất cho mã độc trong quá trình phân tích: Xác định một mẫu malware dựa trên tên tệp là không hiệu quả vì cùng một mẫu malware có thể sử dụng các tên tệp khác nhau, nhưng giá trị băm mã học được tính dựa trên nội dung tệp sẽ giữ nguyên. Do đó, giá trị băm mã hóa cho tệp nghi ngờ của bạn phục vụ như một định danh duy nhất trong quá trình phân tích.

Quyết định liệu phân tích cần được thực hiện trên một mẫu duy nhất hay nhiều mẫu: Trong quá trình phân tích động, khi malware được thực thi, nó có thể sao chép chính nó đến một vị trí khác hoặc tạo ra một mẫu malware khác. Có giá trị băm mã học của mẫu sẽ giúp xác định xem mẫu mới được sao chép/hành động có giống với mẫu gốc hay không. Thông tin này có thể giúp bạn quyết định xem phân tích cần được thực hiện trên một mẫu duy nhất hay nhiều mẫu.

Được sử dụng để chia sẻ cho các nhà nghiên cứu bảo mật khác khi cần xác định mẫu.

Xác định nhanh xem mã độc đã được phát hiện trước đó bằng cách tìm kiếm trực tuyến hoặc tìm kiếm trong cơ sở dữ liệu của các dịch vụ quét mã độc như VirusTotal.

Các công cụ có thể sử dụng:

- Linux: Md5sum, Sha256sum, Sha1sum …
- Windows: HashMyFiles, FsumFrontEnd, Jacksum …

### 3.4.3. Phân tích mẫu đang có sử dụng công cụ kết hợp nhiều Antivirus (AV)

Sử dụng các công cụ tích hợp nhiều AV giúp việc so sánh chữ ký của các mẫu nghi ngờ với cơ sử dữ liệu của các AV. Tên chữ ký cho một tệp cụ thể có thể cung cấp thông tin bổ sung về tệp và khả năng hoạt động của nó. Bằng cách truy cập các trang web của các nhà cung cấp antivirus tương ứng hoặc tìm kiếm chữ ký trên các công cụ tìm kiếm, bạn có thể thu thập thêm thông tin về tệp nghi phạm. Thông tin như vậy có thể giúp trong cuộc điều tra sau này và giảm thời gian phân tích.

VirusTotal là một dịch vụ quét phần mềm độc hại trực tuyến phổ biến. Nó cho phép bạn tải lên một tệp, sau đó tệp này sẽ được quét bằng nhiều chương trình quét antivirus khác nhau và kết quả quét sẽ được hiển thị trực tiếp trên trang web. Ngoài việc tải lên các tệp để quét, giao diện web của VirusTotal cung cấp khả năng tìm kiếm trong cơ sở dữ liệu của họ bằng cách sử dụng giá trị băm (hash), URL, tên miền hoặc địa chỉ IP. VirusTotal cung cấp một tính năng hữu ích khác gọi là VirusTotal Graph, được xây dựng trên cơ sở dữ liệu VirusTotal. Sử dụng VirusTotal Graph, bạn có thể hiển thị mối quan hệ giữa tệp bạn gửi và các chỉ báo kết nối như tên miền, địa chỉ IP và URL. Nó cũng cho phép bạn quay vòng và điều hướng qua mỗi chỉ báo; tính năng này rất hữu ích nếu bạn muốn nhanh chóng xác định các chỉ báo liên quan đến một tệp độc hại. VirusTotal cũng cung cấp khả năng tạo kịch bản thông qua giao diện API công cộng của họ; nó cho phép bạn tự động hóa việc gửi tệp, truy xuất báo cáo quét tệp/URL và truy xuất báo cáo tên miền/IP.

Có một số yếu tố và rủi ro cần xem xét khi quét một tệp nhị phân bằng các chương trình quét Anti-Virus hoặc khi gửi một tệp nhị phân đến các dịch vụ quét Anti-Virus trực tuyến:

Nếu một tệp nhị phân nghi ngờ không bị phát hiện bởi các chương trình quét Anti-Virus, điều đó không nhất thiết có nghĩa rằng tệp đó là an toàn. Các chương trình quét Anti-Virus này dựa vào các chữ ký và phương pháp kỹ thuật để phát hiện các tệp độc hại. Những tác giả phần mềm độc hại có thể dễ dàng sửa mã

của họ và sử dụng các kỹ thuật làm mờ để tránh các phát hiện này, do đó một số chương trình quét Anti-Virus có thể không phát hiện tệp nhị phân nghi phạm là độc hại.

Khi bạn tải lên một tệp nhị phân lên một trang web công cộng, tệp bạn gửi có thể được chia sẻ với bên thứ ba và các nhà cung cấp. Tệp nghi phạm có thể chứa thông tin nhạy cảm, cá nhân hoặc thông tin độc quyền cụ thể cho tổ chức của bạn, vì vậy không nên gửi một tệp nhị phân nằm trong một cuộc điều tra bí mật đến các dịch vụ quét Anti-Virus công cộng. Hầu hết các dịch vụ quét Anti-Virus trực tuyến trên web cho phép bạn tìm kiếm trong cơ sở dữ liệu hiện có của họ bằng cách sử dụng các giá trị băm mã hóa (MD5, SHA1 hoặc SHA256); vì vậy, một cách thay thế để gửi tệp là tìm kiếm dựa trên giá trị băm mã hóa của tệp.

Khi bạn gửi một tệp nhị phân đến các chương trình quét Anti-Virus trực tuyến, kết quả quét được lưu trữ trong cơ sở dữ liệu của họ và hầu hết dữ liệu quét có sẵn công khai và có thể được truy vấn sau này. Các kẻ tấn công có thể sử dụng tính năng tìm kiếm để truy vấn giá trị băm của mẫu của họ để kiểm tra xem tệp nhị phân của họ đã bị phát hiện hay chưa. Việc phát hiện mẫu của họ có thể khiến cho các kẻ tấn công thay đổi chiến thuật của họ để tránh bị phát hiện.

### 3.4.4. Tách chuỗi từ chương trình mã động

Các chuỗi ký tự ASCII và Unicode có thể được tìm thấy trong một tập tin. Việc trích xuất chuỗi có thể cung cấp gợi ý về chức năng của chương trình và các chỉ báo liên quan đến một tập tin nhị phân nghi ngờ. Ví dụ, nếu một phần mềm độc hại tạo một tập tin, tên tập tin đó được lưu trữ dưới dạng một chuỗi trong tập tin nhị phân. Hoặc nếu một phần mềm độc hại giải quyết một tên miền được điều khiển bởi kẻ tấn công, thì tên miền đó được lưu trữ dưới dạng một chuỗi. Các chuỗi được trích xuất từ tập tin nhị phân có thể chứa các tham chiếu đến tên tập tin, URL, tên miền, địa chỉ IP, các lệnh tấn công, các khóa registry, và vân vân. Mặc dù chuỗi không cung cấp một hình ảnh rõ ràng về mục đích và khả năng của một tập tin, chúng có thể đưa ra gợi ý về khả năng của phần mềm độc hại.

**Sử dụng phần mềm để lấy Strings từ các chương trình có thể là mã độc**

Để trích xuất chuỗi từ một tập tin nhị phân nghi ngờ, bạn có thể sử dụng tiện ích strings trên hệ thống Linux. Lệnh strings, mặc định, trích xuất các chuỗi ASCII có độ dài ít nhất là bốn ký tự. Với tùy chọn -a, bạn có thể trích xuất chuỗi từ toàn bộ tập tin.

Đối với các mẫu mã độc hại, cũng sử dụng chuỗi Unicode (2 byte mỗi ký tự). Để có được thông tin hữu ích từ tập tin nhị phân, đôi khi bạn cần trích xuất cả chuỗi ASCII và chuỗi Unicode. Để trích xuất chuỗi Unicode bằng lệnh strings, sử dụng tùy chọn -el.

Trên Windows, PEStudio là một công cụ tiện ích hiển thị cả chuỗi ASCII và chuỗi Unicode. PEStudio là một công cụ phân tích PE tuyệt vời để thực hiện đánh giá ban đầu về phần mềm độc hại từ một tập tin nhị phân nghi ngờ và được thiết kế để thu thập các mẫu thông tin hữu ích từ một tập tin thực thi PE.

**Lấy những chuỗi đã bị che dấu bằng Floss**

Trong hầu hết các trường hợp, tác giả phần mềm độc hại sử dụng các kỹ thuật che dấu chuỗi đơn giản để tránh bị phát hiện. Trong những trường hợp như vậy, những chuỗi đã được che giấu sẽ không xuất hiện trong tiện ích strings và các công cụ trích xuất chuỗi khác. FireEye Labs Obfuscated String Solver (FLOSS) là một công cụ được thiết kế để tự động xác định và trích xuất chuỗi đã được làm mờ từ phần mềm độc hại. Nó có thể giúp bạn xác định những chuỗi mà tác giả phần mềm độc hại muốn che giấu khỏi các công cụ trích xuất chuỗi. FLOSS cũng có thể được sử dụng tương tự như tiện ích strings để trích xuất chuỗi có thể đọc được (ASCII và Unicode).

### 3.4.5. Xác định cách che giấu của tập tin

Dù việc trích xuất chuỗi là một kỹ thuật xuất sắc để thu thập thông tin có giá trị, thường tác giả phần mềm độc hại sẽ che giấu hoặc bảo vệ tập tin nhị phân của họ. Điều này được sử dụng bởi tác giả phần mềm độc hại để bảo vệ cấu trúc bên trong của phần mềm độc hại khỏi các nhà nghiên cứu bảo mật, các chuyên gia phân tích phần mềm độc hại và kỹ sư đảo ngược mã. Các kỹ thuật làm mờ này làm cho việc phát hiện/phân tích tập tin trở nên khó khăn; việc trích xuất chuỗi từ tập tin nhị phân như vậy dẫn đến rất ít chuỗi, và hầu hết các chuỗi được che giấu. Tác giả phần mềm độc hại thường sử dụng các chương trình như Packer và Cryptor để che đi tập tin của họ nhằm tránh sự phát hiện từ các sản phẩm bảo mật như phần mềm chống virus và để ngăn chặn quá trình phân tích.

**Packers and Cryptors**

Packer là một chương trình lấy tập tin thực thi làm đầu vào và sử dụng nén để để che giấu nội dung của tập tin thực thi. Nội dung đã được che giấu sau đó được lưu trữ trong cấu trúc của một tập tin thực thi mới; kết quả là một tập tin thực thi mới (chương trình đã được đóng gói) với nội dung đã được giấu trên ổ đĩa. Khi thực thi chương trình đã được đóng gói, nó thực hiện một quy trình giải nén, trích xuất tập tin nhị phân gốc trong bộ nhớ trong quá trình chạy và kích hoạt việc thực thi. Một Cryptor tương tự như một Packer, nhưng thay vì sử dụng nén, nó sử dụng mã hóa

để giấu nội dung của tập tin thực thi, và nội dung đã được mã hóa được lưu trữ trong tập tin thực thi mới. Khi thực thi chương trình đã được mã hóa, nó chạy một quy trình giải mã để trích xuất tập tin nhị phân gốc trong bộ nhớ và sau đó kích hoạt việc thực thi.

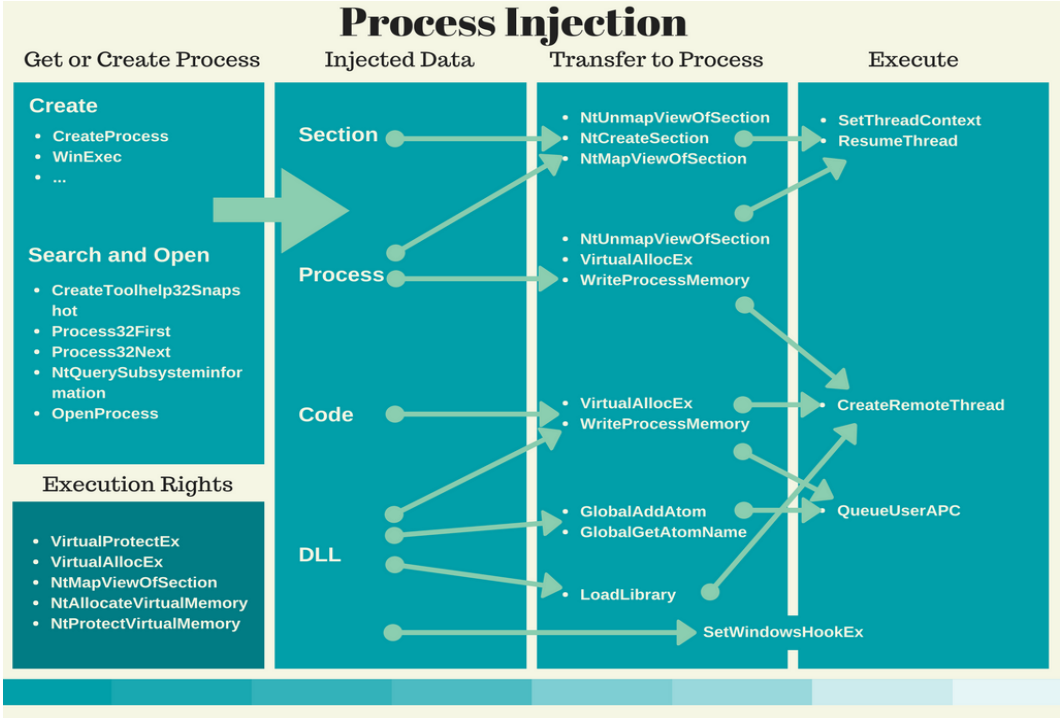
**Phát hiện các tệp tin ẩn giấu bằng Exeinfo PE**

Hầu hết các tập tin thực thi hợp pháp không che giấu nội dung, nhưng một số tập tin thực thi có thể thực hiện điều này để ngăn người khác kiểm tra mã nguồn của họ. Khi bạn gặp một mẫu mã đã được đóng gói, có khả năng cao nó sẽ là độc hại. Để phát hiện các chương trình đóng gói trên Windows, bạn có thể sử dụng một công cụ miễn phí như Exeinfo PE ,nó có giao diện người dùng dễ sử dụng, để phát hiện các trình biên dịch, chương trình đóng gói hoặc mã hóa được sử dụng để xây dựng chương trình.

### 3.4.6. Nhận biết mã độc có bị nén hay mã hóa không?

Các công cụ nhận dạng gói (packers) đã biết sẽ được xác định bằng các nhận dạng gói như Detect it Easy, PEiD, và các công cụ tương tự. Tuy nhiên, phần mềm độc hại thường được gói bởi các gói tùy chỉnh, được tạo ra đặc biệt để tránh các chương trình antivirus. Những gói này không thể được xác định bằng những công cụ này. Có một số nguyên tắc tổng quát có thể chỉ ra liệu một tập tin có được gói hay không, như:

- Có độ entropy cao
- Các hàm nhập điển hình của packed files hay nhiễm độc




Có dòng rác, không có nghĩa trong tệp nhị phân

### 3.4.7. Tệp tin PE

Các tệp thực thi Windows (như .exe, .dll, .sys, .ocx và .drv) phải tuân thủ theo định dạng PE (Portable Executable).

Tệp PE là một loạt các cấu trúc và thành phần con chứa thông tin cần thiết cho hệ điều hành để tải nó vào bộ nhớ.

Khi một tệp thực thi được biên dịch, nó bao gồm header (PE header), mô tả cấu trúc của nó. Khi tệp nhị phân được thực thi, trình nạp hệ điều hành đọc thông tin từ PE header và sau đó nạp nội dung nhị phân từ tệp vào bộ nhớ. PE header chứa thông tin như nơi tệp thực thi cần được nạp vào bộ nhớ, địa chỉ mà quá trình thực thi bắt đầu, danh sách thư viện/chức năng mà ứng dụng phụ thuộc vào, và tài nguyên được sử dụng bởi tệp nhị phân. Kiểm tra PE header mang lại một lượng thông tin lớn về tệp nhị phân và tính năng của nó.

Công cụ:

- CFF Explorer: http://www.ntcore.com/exsuite.php
- PEInternals:http://www.andreybazhan.com/pe-internals.html
- PPEE(puppy): https://www.mzrst.com

Thường, phần mềm độc hại tương tác với tệp, registry, mạng, và các thành phần khác. Để thực hiện các tương tác như vậy, phần mềm độc hại thường phụ thuộc vào các hàm được hệ điều hành cung cấp. Windows xuất ra hầu hết các hàm của nó, gọi là Application Programming Interfaces (API), cần thiết cho các tương tác này trong các tệp Dynamic Link Library (DLL). Các tệp thực thi nhập và gọi các hàm này thường từ các DLL khác nhau cung cấp các chức năng khác nhau. Các hàm mà một tệp thực thi nhập từ các tệp khác (chủ yếu là DLL) được gọi là các hàm nhập (hàm imports).VD: Nếu mã độc muốn tạo một file trên ổ đĩa Windows nó cần sử dụng API CreateFile(), được cung cấp bởi kernel32.dll, nó cần tải kernel32.dll vào bộ nhớ và sau đó gọi hàm CreateFile().

Kiểm tra các imports có thể:

- Cung cấp một thông tin về chức năng và khả năng của mã độc và giúp dự đoán những hoạt động của nó trong quá trình thực thi.
- Xác định xem mã độc có được che giấu hay không

Kiểm tra Exports

- Để đánh giá chức năng của DLL
- Kẻ tấn công có thể tạo ra các DLL có các chức năng chứa các chức năng của mã độc và thường được sử dụng bởi chương trình khác

Nội dung của tệp PE được chia thành các Section. Các section này đại diện cho code hoặc data và chúng có các thuộc tính trong bộ nhớ như read/ write . Phần đại diện cho code chứa các hướng dẫn sẽ được thực thi bởi bộ xử lý, trong khi phần chứa dữ liệu có thể đại diện cho các loại dữ liệu khác nhau, chẳng hạn như dữ liệu chương trình đọc / viết (biến toàn cục), bảng import/export, tài nguyên, và cùng một số khác. Mỗi section có tên riêng biệt thể hiện mục đích của section. Ví dụ, một section với tên là .text chỉ ra code và có thuộc tính read-execute; một section với tên .data chỉ ra dữ liệu toàn cầu và có thuộc tính read-write.

Trong quá trình biên dịch của tệp thực thi, các trình biên dịch thêm các tên section liên tục. Bảng sau đây cho biết một số section thông thường trong một tệp PE:

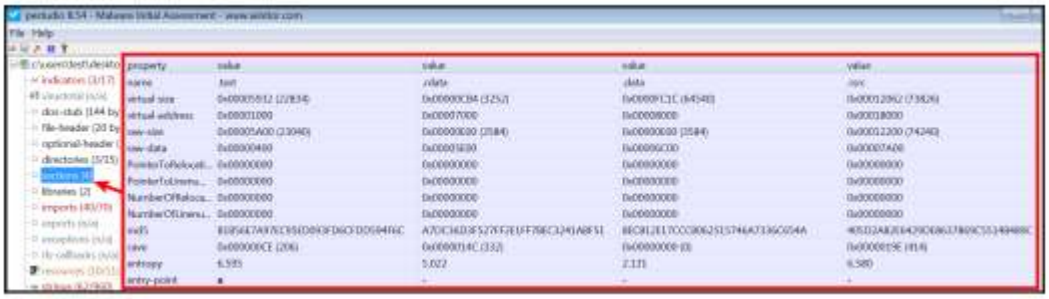
| Section Name | Description |
|---|---|
| `.text` or `CODE` | Contains executable code. |
| `.data` or `DATA` | Typically Contains read/write data and global variables. |
| `.rdata` | Contains read-only data. Sometimes it also contains import and export information. |
| `.idata` | If present, contains the import table. If not present, then the import information is stored in `.rdata` section. |
| `.edata` | If present, contains export information. If not present, then the export information is found in `.rdata` section. |
| `.rsrc` | This section contains the resources used by the executable such as icons, dialogs, menus, strings, and so on. |

Các tên section này chủ yếu dành cho con người và không được hệ điều hành sử dụng, điều này có nghĩa rằng có khả năng một kẻ tấn công hoặc phần mềm làm mờ thông tin có thể tạo ra các section với tên khác nhau. Nếu bạn gặp các tên section không phổ biến, thì bạn nên xem xét chúng với sự nghi ngờ, và cần thêm phân tích để xác minh tính độc hại.

Thông tin về các section này (như tên phần, vị trí của phần và các đặc điểm của nó) có sẵn trong bảng phần trong PE header (Portable Executable). Kiểm tra bảng phần sẽ cung cấp thông tin về phần và các đặc điểm của nó.

Khi bạn tải một tệp thực thi trong pestudio và nhấp vào mục "sections," nó sẽ hiển thị thông tin về các phần được trích xuất từ bảng phần và các thuộc tính của chúng.

| Field | Description |
|---|---|
| Names | Displays section names. In this case, the executable contains four sections (`.text`, `.data`, `.rdata` and `.rsrc`). |
| Virtual-Size | Indicates the size of the section when loaded into memory. |
| Virtual-Address | This is the relative virtual address (that is, offset from the base address of the executable) where the section can be found in memory. |
| Raw-size | Indicates the size of the section on the disk. |
| Raw-data | Indicates the offset in the file where the section can be found. |
| Entry-point | This is the RVA (relative virtual address) where the code starts executing. In this case, the entry point is in the `.text` section, which is normal. |



Kiểm tra PE Resources: Các tài nguyên cần thiết bởi tệp thực thi như biểu tượng, menu, hộp thoại và chuỗi được lưu trữ trong phần tài nguyên (.rsrc) của tệp thực thi. Thường, các kẻ tấn công lưu trữ thông tin như các tệp nhị phân bổ sung, tài liệu mạo danh và dữ liệu cấu hình trong phần tài nguyên, vì vậy kiểm tra tài nguyên có thể tiết lộ thông tin quý giá về một tệp thực thi. Phần tài nguyên cũng chứa thông tin về phiên bản có thể tiết lộ thông tin về nguồn gốc, tên công ty, thông tin tác giả chương trình và thông tin bản quyền. Resource Hacker là một công cụ tuyệt vời để kiểm tra, xem và trích xuất tài nguyên từ một tệp thực thi nghi ngờ.

### 3.4.8. Phân loại mã độc

Phân tích mẫu mã độc có thể cho biết mẫu đó có thuộc họ mã độc (family), hoặc có tính chất giống với những mẫu đã phân tích trước đó hay không. Cụ thể:

- Phân tích mẫu mã độc có thể xác định nó thuộc gia đình malware nào, chẳng hạn như ransomware, trojan, backdoor, ...

- So sánh mẫu đó với các mẫu đã biết để xác định nó có đặc trưng giống với mẫu đã biết trước đó hay không, ví dụ so sánh mã, hành vi, kỹ thuật lây lan, ...
- Nếu so sánh có kết quả gần giống thì có thể kết luận đó là một mẫu thuộc cùng gia đình (family) hoặc biến thể của một mã đã biết

Mặc dù việc tạo băm mã hóa (MD5/SHA1/SHA256) là một kỹ thuật tốt để phát hiện các mẫu giống nhau, nhưng nó không giúp xác định các mẫu tương tự. Rất nhiều lần, tác giả phần mềm độc hại thay đổi những khía cạnh rất nhỏ của phần mềm độc hại, điều này thay đổi hoàn toàn giá trị băm.

Một số kỹ thuật có thể giúp so sánh và phân loại tệp đang nghi ngờ:

- Phân loại mã độc sử dụng Fuzzy Hashing:

  Fuzzy hashing là một phương pháp để so sánh sự tương đồng giữa các tập tin.

  Kỹ thuật này so sánh một nhị phân nghi ngờ với các mẫu trong kho lưu trữ để xác định các mẫu tương tự là xác định các mẫu thuộc cùng gia đình malware hoặc cùng nhóm tác nhân

- Phân loại mã độc sử dụng Import Hash (imphash):

  Phân tích mã độc sử dụng các hàm API (hay còn gọi là imphash) là một kỹ thuật có thể được sử dụng để xác định mẫu liên quan và mẫu được sử dụng bởi các nhóm tác nhân đe dọa.

  Đặc điểm này dựa trên việc tính toán giá trị băm dựa trên tên hàm/ hàm nhập (API) và thứ tự của chúng trong tập tin thực thi. Nếu các tập tin được biên dịch từ cùng một nguồn và theo cùng một cách, những tập tin đó sẽ có giá trị imphash giống nhau.

  Trong quá trình điều tra mã độc, nếu phát hiện các mẫu có cùng giá trị imphash, điều đó có nghĩa chúng có cùng bảng địa chỉ nhập và có khả năng liên quan đến nhau.

  ➢ Phân loại mã độc sử dụng Section Hash

Tương tự như Import Hash, Section Hash cũng giúp xác định các mã độc có liên quan đến nhau.

➢ Phân loại mã độc sử dụng Yara.

YARA là một công cụ mạnh mẽ để nhận diện và phân loại mã độc.

Có thể tạo các quy tắc YARA dựa trên thông tin văn bản hoặc nhị phân có trong mẫu mã độc.

Các quy tắc YARA này bao gồm một tập hợp các chuỗi và một biểu thức logic. Những quy tắc sau khi được viết có thể dùng để quét tệp tin bằng công cụ YARA hoặc có thể sử dụng yara python để tích hợp với các công cụ khác.

- Quy tắc YARA bao gồm các thành phần sau:

Rule identifier (Định danh quy tắc): Đây là tên mô tả quy tắc. Các định danh quy tắc có thể chứa bất kỳ ký tự chữ và số nào cùng với ký tự gạch dưới, nhưng ký tự đầu tiên không thể là một chữ số. Các định danh quy tắc phân biệt chữ hoa chữ thường và không vượt quá 128 ký tự.

String Definition (Định nghĩa chuỗi): Đây là phần mà các chuỗi sẽ được định nghĩa và sử dụng trong quy tắc. Phần này có thể bị bỏ qua nếu quy tắc không phụ thuộc vào bất kỳ chuỗi nào. Mỗi chuỗi có một định danh được tạo bởi ký tự $ theo sau là một chuỗi gồm các ký tự chữ và số cùng với ký tự gạch dưới.

Condition Section (Phần điều kiện): là nơi logic của quy tắc được đặt. Phần này phải chứa một biểu thức Boolean xác định điều kiện mà quy tắc sẽ khớp hoặc không khớp.

### 3.5. Đánh giá

# II. CÀI ĐẶT & CẤU HÌNH

1. **Cài đặt**

Đường dẫn tải các công cụ và malware:

- **PEStudio**: https://pestudio.en.lo4d.com/download/mirror-ls1
- **Cutter**: https://cutter.re/
- **Cmder**: https://cmder.app/
- **Capa**:https://github.com/mandiant/capa/releases
- **Floss**: https://github.com/mandiant/flare-floss/releases
- **CFF Explorer**: https://ntcore.com/?page_id=388

2. **Cấu hình cài đặt:**

**2.1. Giải nén trực tiếp và chạy công cụ.**

Sau khi download thành công PeStudio, Cutter, CFF Explorer tiến hành giải nén file, và chạy công cụ.

**2.2. Cấu hình một số tool**

Sau khi download thành công Cmder, tiến hành giải nén và chạy Cmder.

Tại đây bắt đầu cấu hình cài đặt Capa và Floss theo đường dẫn cài đặt

# III. TROJAN ZEUS BANKING.

**Zeus Banking Trojan** hay còn gọi là ZueS hoặc Zbot, là một gói phần mềm mãi độc dạng Trojan, nó phổ biến và đáng ngại trong lĩnh vực tài chính trên internet. Nó được thiết kế để đánh cắp thông tin đăng nhập của người dùng, cho phép kẻ tấn công truy cập và lừa đảo trong quá trình giao dịch tài khoản trực tuyến.

Mã độc này hoạt động như một phần mềm độc hại có khả năng đánh cắp thông tin và hoạt động trong các môi trường trực tuyến, chủ yếu là tập trung vào việc tấn công hệ thống ngân hàng và tài chính. Nó có thể ghi lại thông tin đăng nhập, số thẻ tín dụng, mật khẩu và các thông tin nhạy cảm từ máy tính của nạn nhân.

Zeus Banking Trojan thường lây lan qua các email lừa đảo (Phishing) hoặc các trang web giả mạo. Khi người dùng bấm vào liên kết hoặc tải xuống tệp đính kèm trong email hoặc trang web độc hại, mã độc sẽ tự động cài đặt và thực hiện trên máy tính của người dùng mà không cần sự cho phép.

Nó có khả năng gửi các thông tin đăng nhập và dữ liệu nhạy cảm từ máy tính của nạn nhân về một máy chủ điều khiển từ xa (command-and-control server) được kiểm soát bởi kẻ tấn công. Kẻ tấn công sau đó có thể sử dụng thông tin này để đánh cắp tiền

từ tài khoản ngân hàng, thực hiện giao dịch giả mạo hoặc tiếp tục các hoạt động lừa đảo khác.

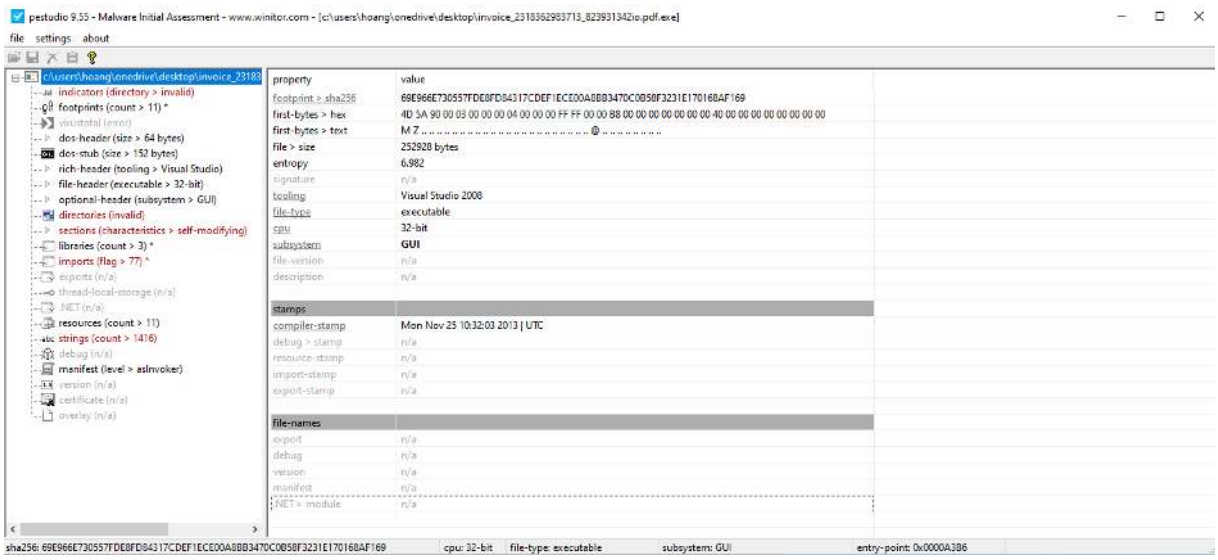
Zeus Banking Trojan đã xuất hiện từ những năm 2007 và đã trải qua sự phát triển và tiến hóa liên tục. Nó đã tạo ra nhiều biến thể và phiên bản khác nhau, nhằm tránh phát hiện và phòng ngừa từ các giải pháp bảo mật. Các biến thể của Zeus Banking Trojan có thể có tính năng bổ sung và phương thức tấn công khác nhau.

Nó gây ra nguy cơ lớn đối với người dùng và tổ chức tài chính. Nó có thể gây ra mất cắp tài sản, mất thông tin cá nhân, và tiềm tàng nguy cơ cho sự riêng tư và an ninh tài chính của người dùng. Nó cũng có thể gây ra thiệt hại danh tiếng và kinh tế đối với các tổ chức bị tấn công.

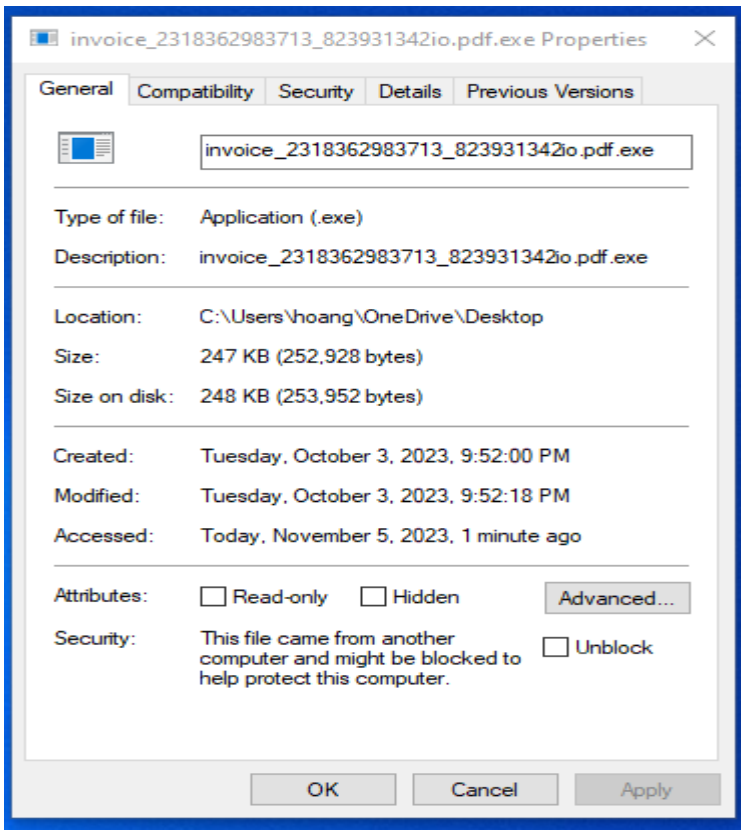
**TrojanBankingMalware**:https://github.com/ytisf/theZoo/tree/master/malware/Binaries/ZeusBankingVersion_26Nov2013

# IV. PHÂN TÍCH - THỰC NGHIỆM

## 1. Xác định loại của tệp



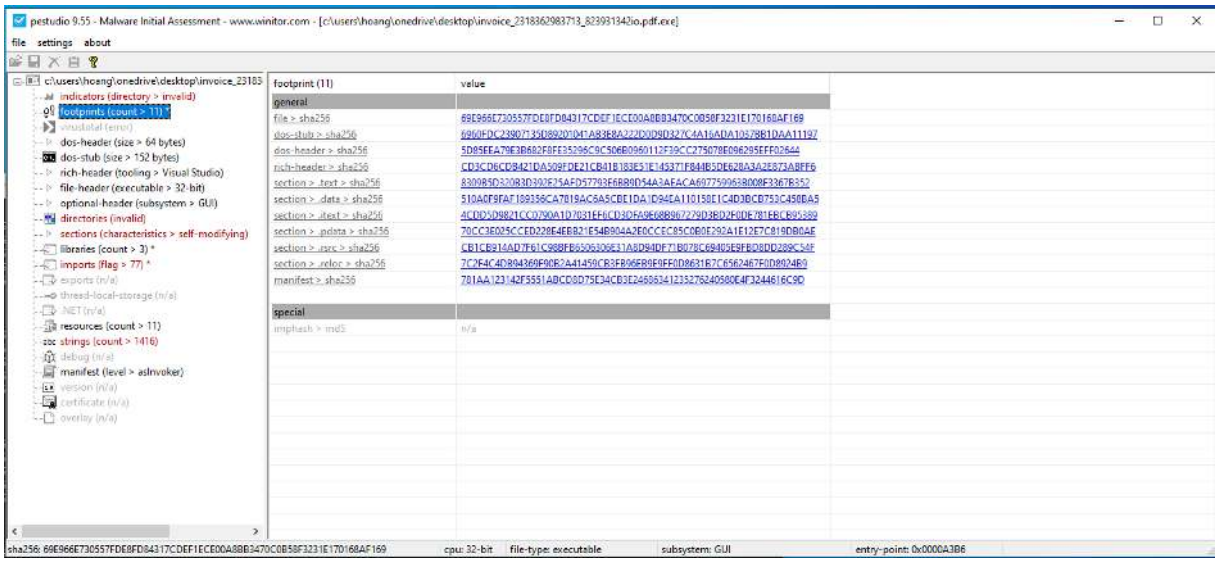

- Ta dùng PEStudio để phân tích file nghi ngờ. Tại first-byte > text , ta xác định được chữ ký tệp là MZ. Kích thước tệp là 252928 bytes.
- Ta thấy file-type là Portable Executable 32-bit tức là file này được thiết kế để tấn công hệ điều hành Windows.
- Tại phần Stamps ta xác định được thời điểm file được biên dịch lần đầu vào Mon Nov 25 10:32:03 2013.



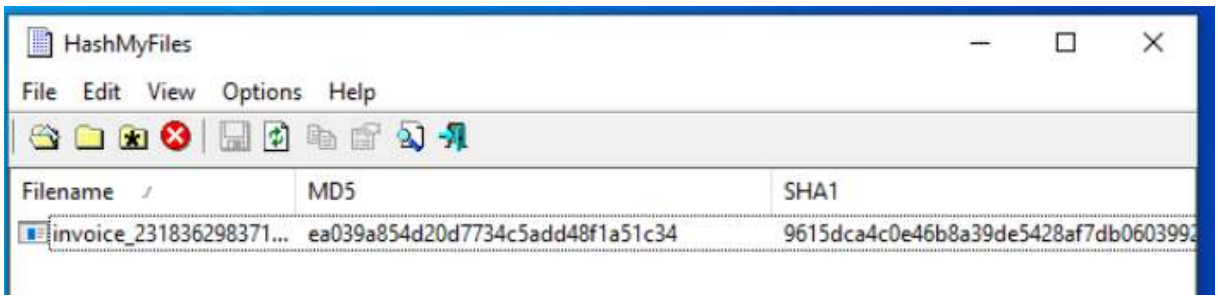

- Ban đầu ta thấy file có định dạng pdf nhưng để chắc chắn hơn ta kiểm tra lại. Vào properties thì ta thấy tên file là ***invoice_2318362983713_823931342io.pdf.exe*** . Như có thể thấy định dạng của file là exe nhưng nó đang cố giả dạng thành file pdf để đánh lừa người dùng.

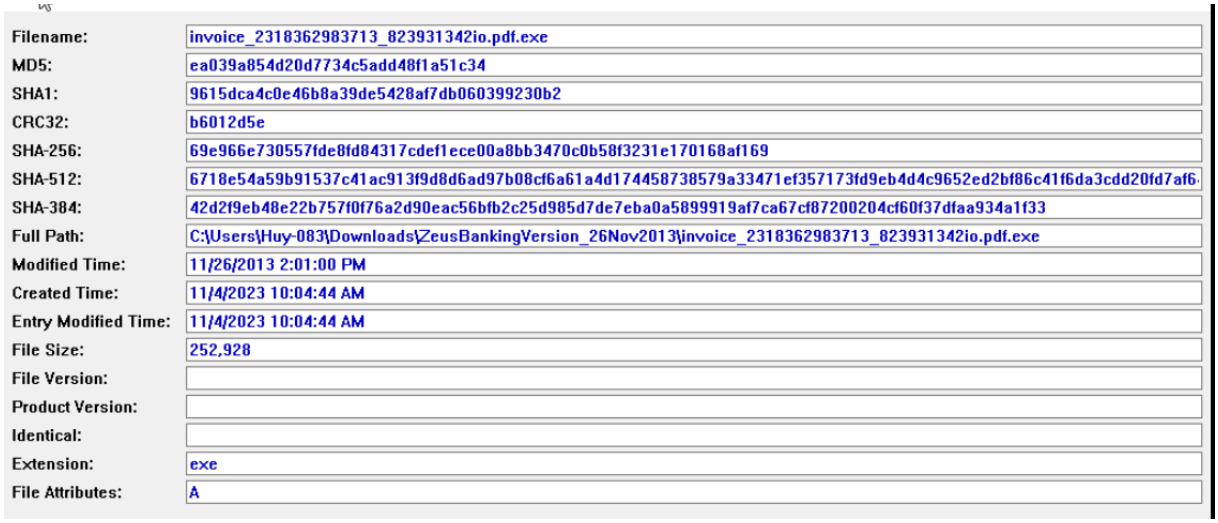
## 2. Phân tích mã HASH

### 2.1. Kiểm Tra mã HASH thu được của Pestudio



### 2.2. Sử dụng HashMyFiles để tính toán mã Hash của File gốc




### 2.3. So sánh - Kiểm Tra tính toàn vẹn của file mã độc

So sánh mã hash thu được khi phân tích mã độc bằng Pestudio và mã độc tính toán được từ HashMyFiles,

*69E966E730557FDE8FD84317CDEF1ECE00A8BB3470C0B58F3231E170168AF169 (Pestudio)*

*69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169 (HashMyFile)*

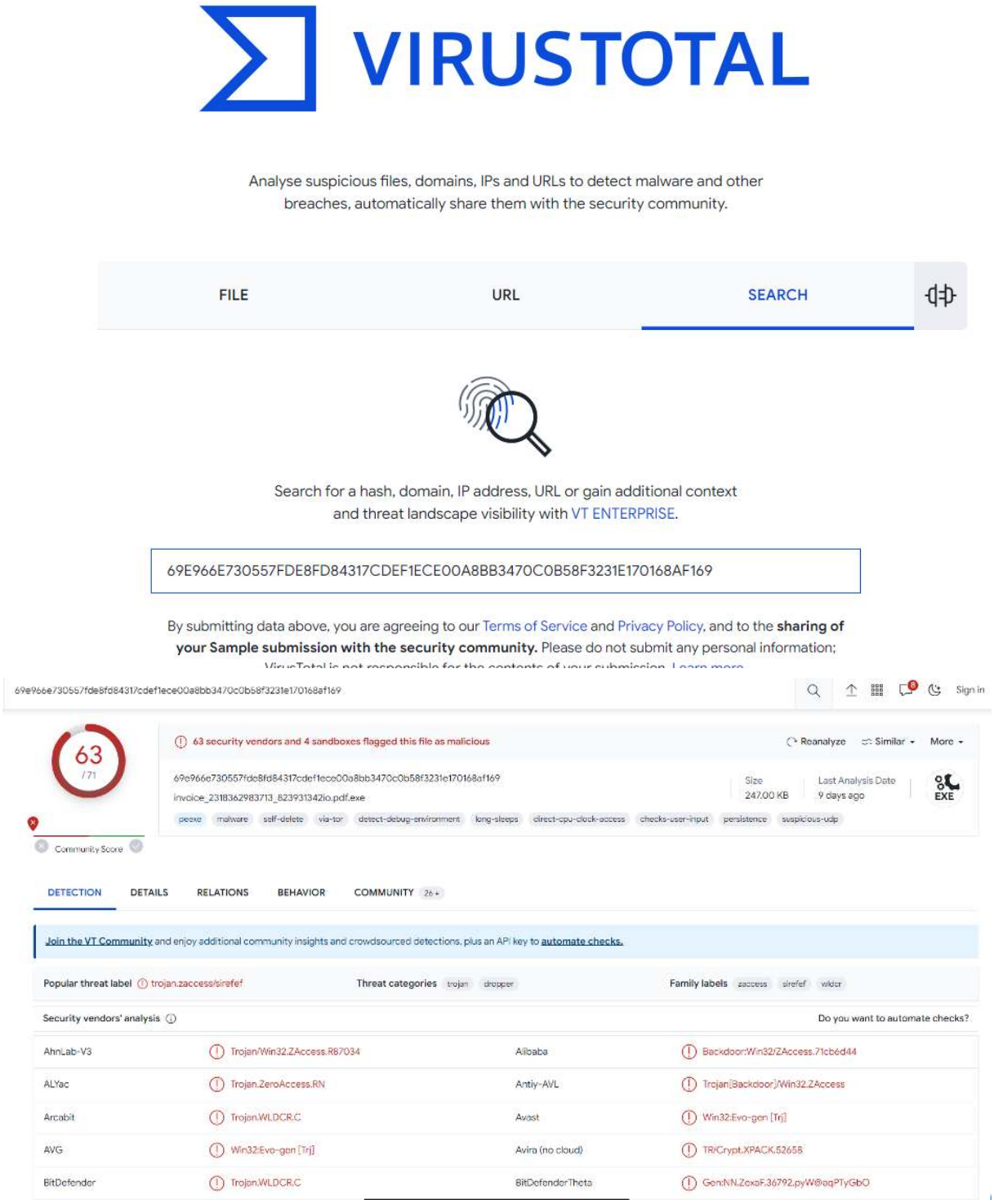
=> Hai mã có sự tương đồng, điều này cho thấy File không bị thay đổi mã từ lúc bắt đầu tính toán mã Hash => Tính toàn vẹn

Việc xác định tính toàn vẹn này là cần thiết vì trong thực tế Phân Tích Tĩnh và Phân Tích Động thường được sử dụng kết hợp với nhau.

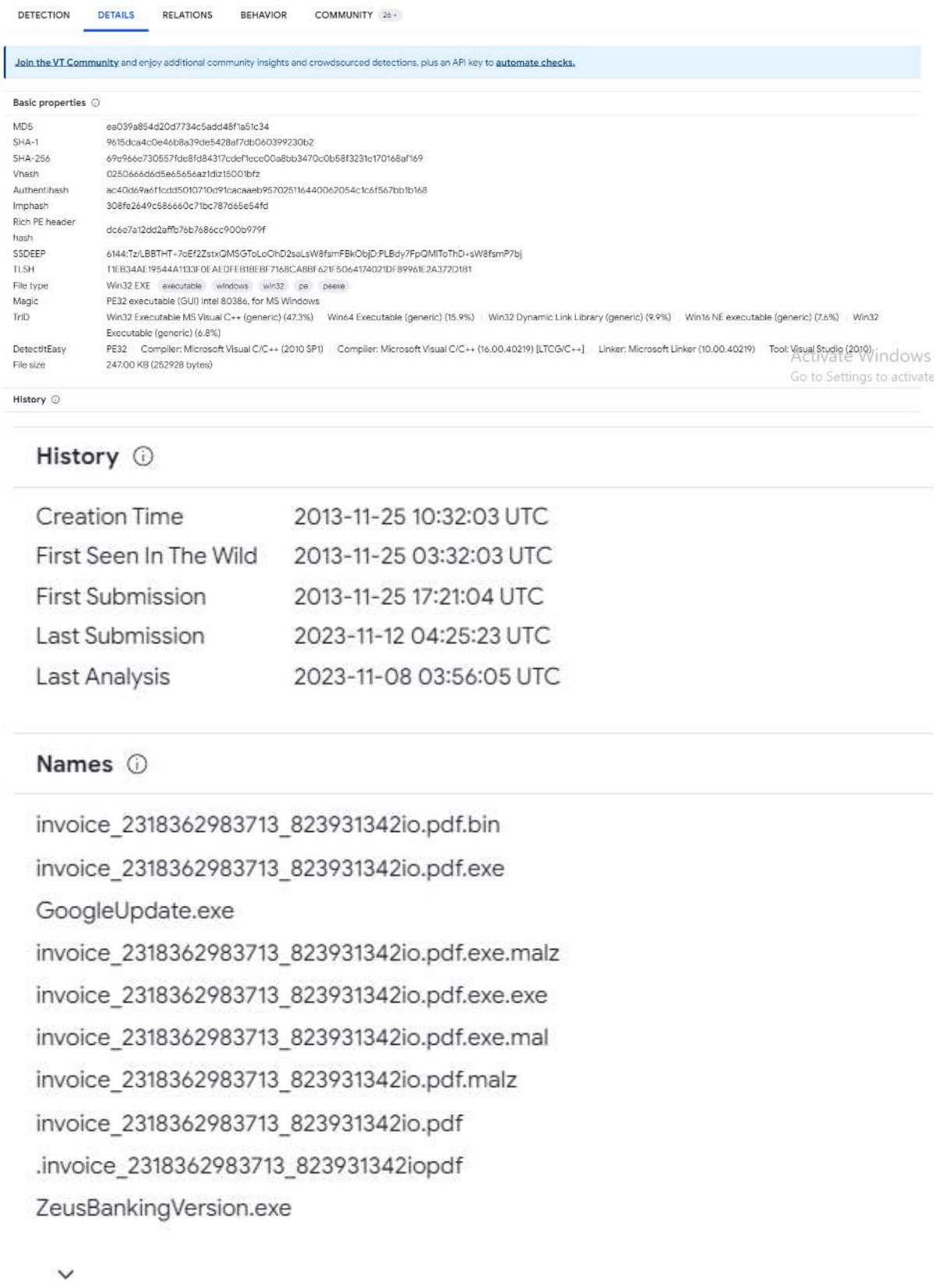
Trong quá trình phân tích động, khi malware được thực thi, nó có thể sao chép chính nó đến một vị trí khác hoặc tạo ra một mẫu malware khác. Có giá trị băm mã học của mẫu sẽ giúp xác định xem mẫu mới được sao chép/hành động có giống với mẫu gốc hay không.

=> Trong trường hợp này ta có thể thấy được Mã Hash của file mã độc không hề thay đổi trong quá trình phân tích => Giúp ta biết được nên phân tích 1 hay nhiều mẫu

## 3. Phân tích mẫu đang có sử dụng công cụ kết hợp nhiều Antivirus (AV)




- Như ta có thể thấy có 63 nhà cung cấp cho rằng file có hại.



**Portable Executable Info** ⓘ

**Compiler Products**

[ASM] VS2008 SP1 build 30729 count=1
[IMP] VS2008 SP1 build 30729 count=11
[---] Unmarked objects count=86
id: 175, version: 40219 count=19
[EXP] VS2010 SP1 build 40219 count=1
[RES] VS2010 SP1 build 40219 count=1
[LNK] VS2010 SP1 build 40219 count=1

**Header**

| | |
|---|---|
| Target Machine | Intel 386 or later processors and compatible processors |
| Compilation Timestamp | 2013-11-25 10:32:03 UTC |
| Entry Point | 41910 |
| Contained Sections | 6 |

**Sections**

| Name | Virtual Address | Virtual Size | Raw Size | Entropy | MD5 | Chi2 |
|---|---|---|---|---|---|---|
| .text | 4096 | 46449 | 46592 | 6.71 | 679fbf23d7317d8207d350b532908f0a | 289621.75 |
| .data | 53248 | 75953 | 76288 | 6.13 | 73fdae90c1738941b6afec633c45972e | 485543.81 |
| .itext | 131072 | 2125 | 2560 | 4.82 | 7f89ad170ffea80a9c7304edf9c7f32c | 74981.61 |
| .pdata | 135168 | 97470 | 97792 | 6.77 | a8448d1b94e56bc8f80ed852445884c1 | 183410.42 |
| .rsrc | 233472 | 22770 | 23040 | 6.14 | b3af18982aee2e1b39915237800c877e | 137505.08 |

**Imports**

+ KERNEL32.dll
+ SHLWAPI.dll
+ USER32.dll

**Contained Resources By Type**

RT_CURSOR 10

**Contained Resources By Language**

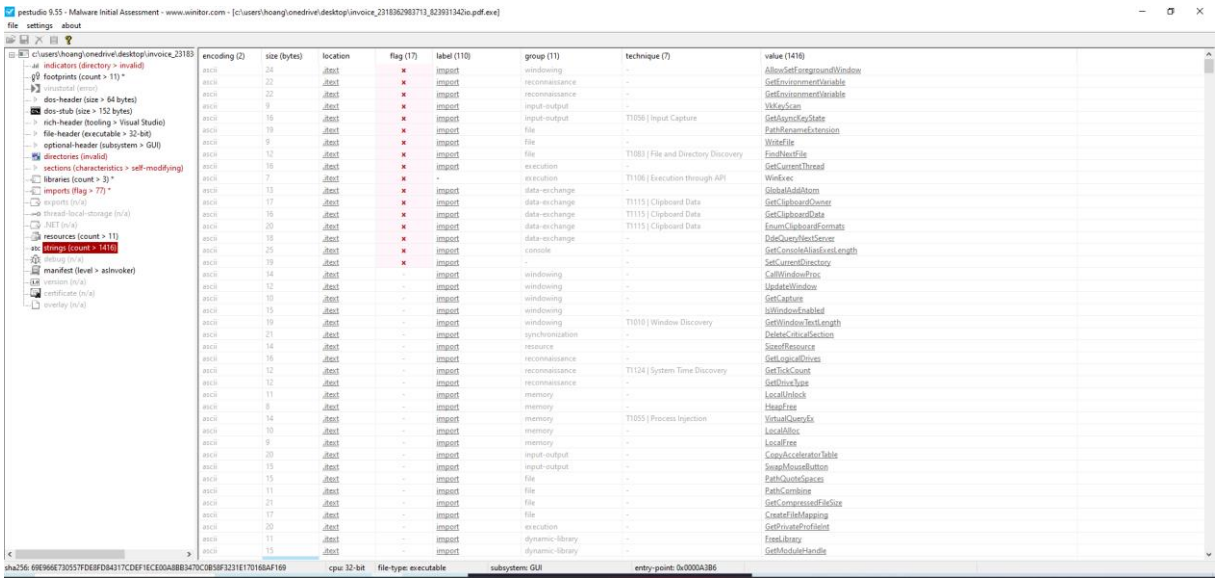
ENGLISH US 10

**Contained Resources**

| SHA-256 | File Type | Type | Language | Entropy | Chi2 |
|---|---|---|---|---|---|
| d16647118d8626a75a2a914a90595f7f2d6606a6cb0f2cfacfe9d55c58f6ef2f | unknown | RT_CURSOR | ENGLISH US | 3.86 | 81359.25 |
| 4417ad2c621b700b3d3ea2c7436f7940b67346aa7eb721aa47bf7e01e9931ac1 | unknown | RT_CURSOR | ENGLISH US | 4.11 | 88287.31 |
| d615652e1b7046649c5189bda030419f39e76fdf4041c7a6821988d773ec5a05 | unknown | RT_CURSOR | ENGLISH US | 4.16 | 101914.94 |
| 17a64d89b34cce3d24d08337d76af5abb7650bb2203c9f3d5039786f89dd2a8d | unknown | RT_CURSOR | ENGLISH US | 3.9 | 88566.18 |
| f3bcf4767282b3d10d915f0c33e7372eed0782679a641dd00caa9ee66984a80a | unknown | RT_CURSOR | ENGLISH US | 3.99 | 84684.11 |

# 4. Tách chuỗi từ chương trình

## 4.1. Sử dụng Pestudio để tách chuỗi từ chương trình



Sử dụng PeStudio để hiển thị cả chuỗi ASCII và chuỗi Unicode.

Có tổng cộng trên 1416 chuỗi trong tập tin, nhưng PEStudio giới hạn số chuỗi nên chỉ hiển thị 1416 chuỗi đầu tiên

Trong công  cụ PeStudio, các chuỗi đánh dấu cờ "x" thường cho rằng chúng có thể liên quan đến hành vi độc hại của mã độc.

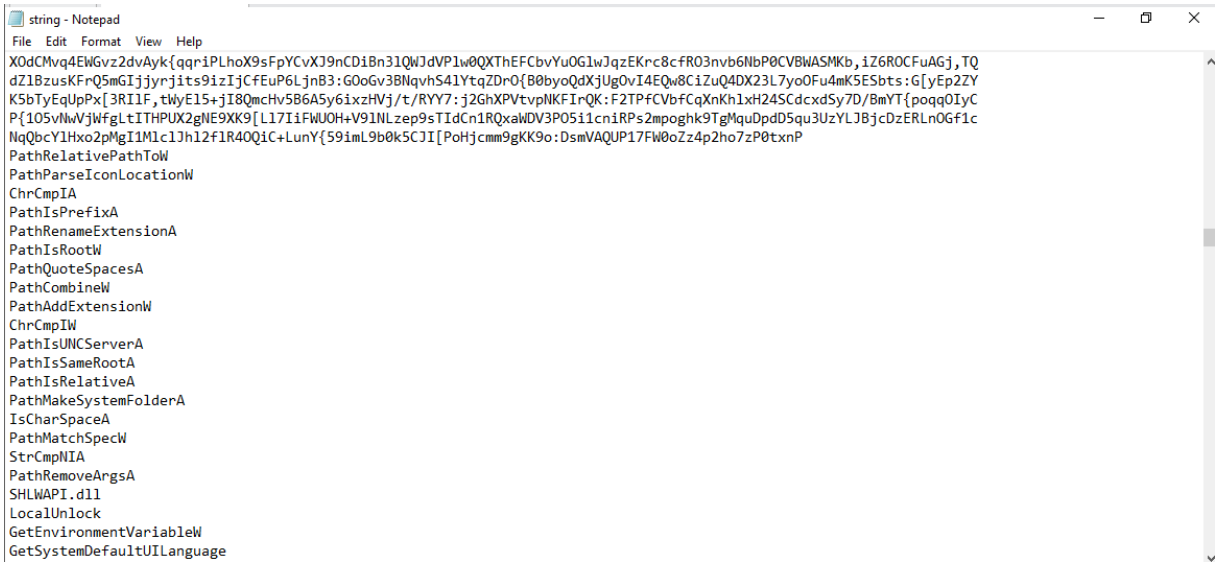
### 4.2. Lấy chuỗi bị che dấu bằng floss

- Sẽ có những chuỗi mà mã độc không muốn ta phát hiện ra nên đã che giấu đi và để lấy được những chuỗi được che giấu đó ta sử dụng FLOSS.

strings.txt - Notepad

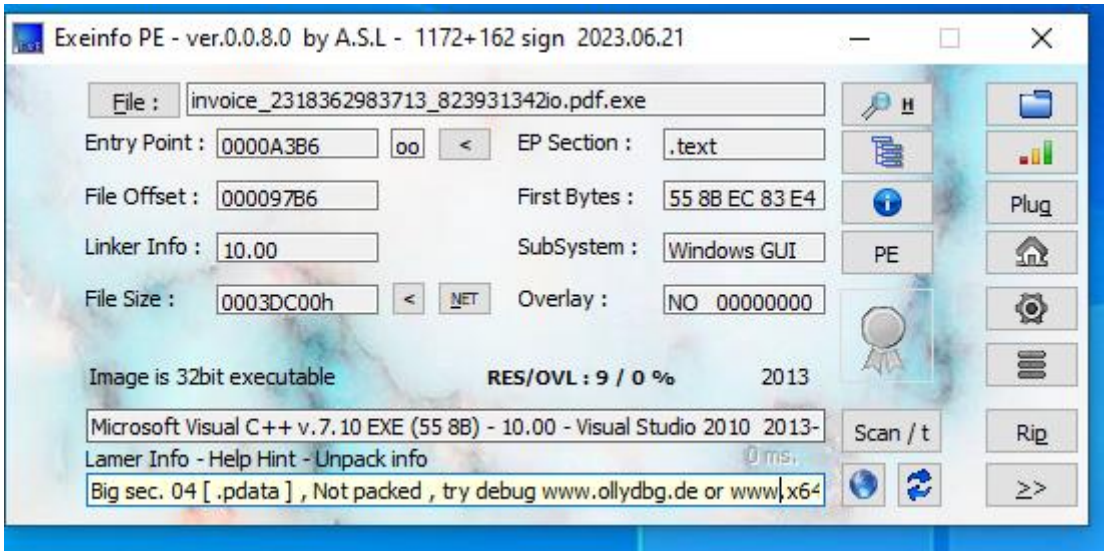
File Edit Format View Help

```
FLARE FLOSS RESULTS (version quantumstrand-preview6-0-g906f5fa)

+-----------------------+------------------------------------------------------------------------------+
| file path             | invoice_2318362983713_823931342io.pdf.exe                                    |
| extracted strings     |                                                                              |
|  static strings       | 791                                                                          |
|  stack strings        | 2                                                                            |
|  tight strings        | 0                                                                            |
|  decoded strings      | 66                                                                           |
+-----------------------+------------------------------------------------------------------------------+


 FLOSS STATIC STRINGS

+-----------------------------------+
| FLOSS STATIC STRINGS: ASCII (790) |
+-----------------------------------+

!This program cannot be run in DOS mode.
Rich
.text
`.data
.itext
```



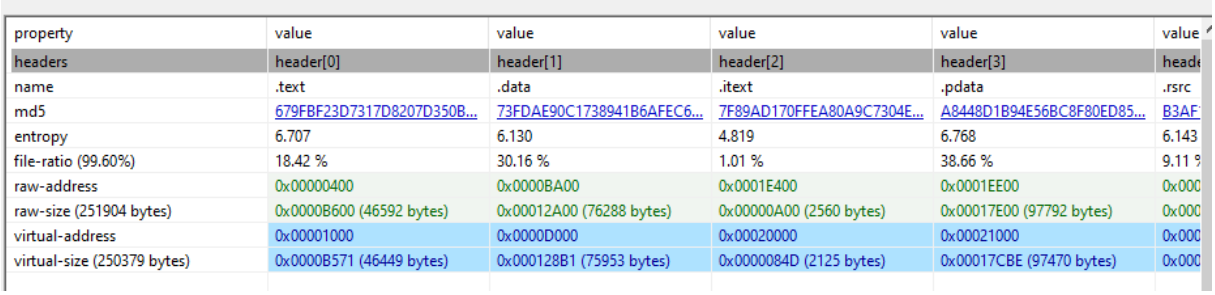

## 5. Xác định các cách che giấu của tập tin

### 5.1. Xác định các tệp tin ẩn giấu bằng Exinfo PE




- Exeinfo PE giúp ta thấy được file không hề bị đóng gói.

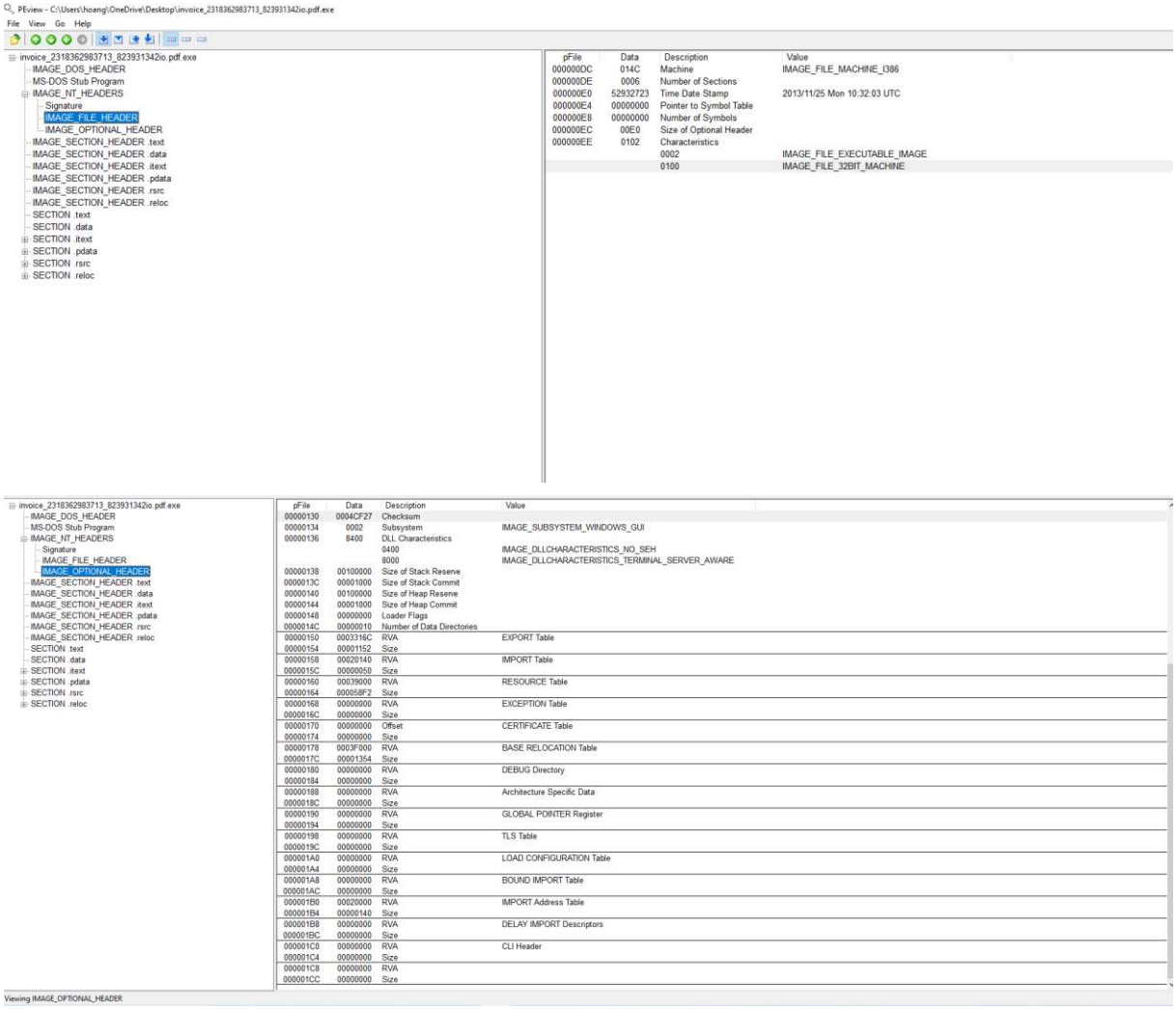
## 6. Nhận biết mã độc có bị nén hay mã hóa hay không



| property | value | value | value | value | value |
|---|---|---|---|---|---|
| headers | header[0] | header[1] | header[2] | header[3] | head |
| name | .text | .data | .itext | .pdata | .rsrc |
| md5 | 679FBF23D7317D8207D350B... | 73FDAE90C1738941B6AFEC6... | 7F89AD170FFEA80A9C7304E... | A8448D1B94E56BC8F80ED85... | B3AF |
| entropy | 6.707 | 6.130 | 4.819 | 6.768 | 6.143 |
| file-ratio (99.60%) | 18.42 % | 30.16 % | 1.01 % | 38.66 % | 9.11 % |
| raw-address | 0x00000400 | 0x0000BA00 | 0x0001E400 | 0x0001EE00 | 0x000 |
| raw-size (251904 bytes) | 0x0000B600 (46592 bytes) | 0x00012A00 (76288 bytes) | 0x00000A00 (2560 bytes) | 0x00017E00 (97792 bytes) | 0x000 |
| virtual-address | 0x00001000 | 0x0000D000 | 0x00020000 | 0x00021000 | 0x000 |
| virtual-size (250379 bytes) | 0x0000B571 (46449 bytes) | 0x000128B1 (75953 bytes) | 0x0000084D (2125 bytes) | 0x00017CBE (97470 bytes) | 0x000 |

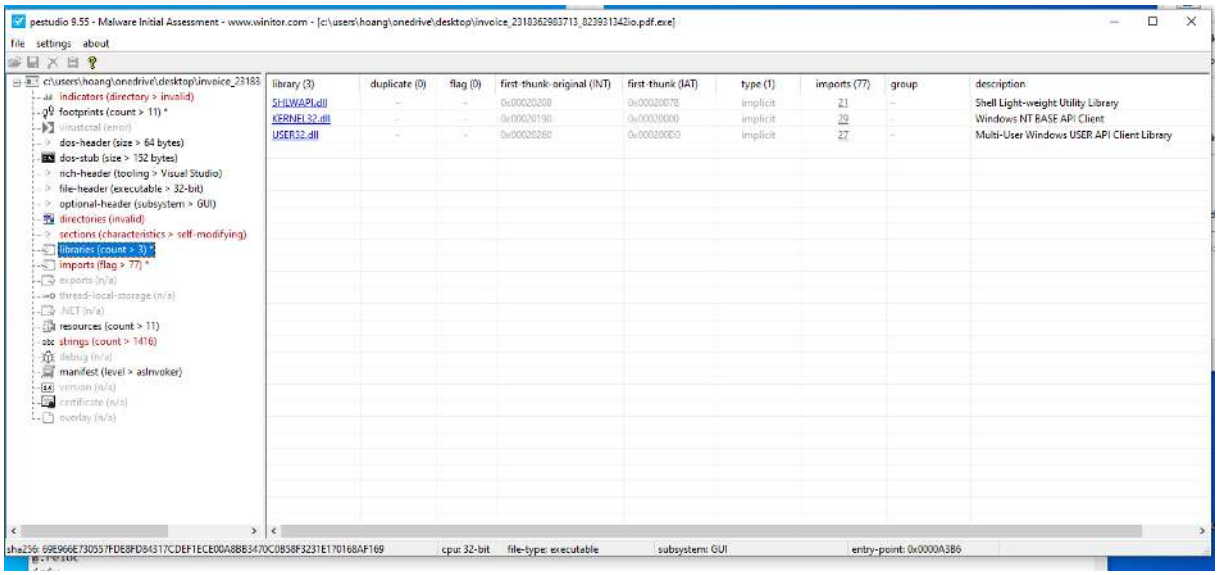
- Ta kiểm tra kích thước của raw-size và virtual-size. Nếu kích thước chênh lệch quá lớn thì có khả năng nó đăng được đóng gói hoặc mã hóa. Nhưng ở đây ta thấy độ chênh lệch không quá lớn nên không có đóng gói hoặc mã hóa.

## 7. Tệp tin PE




+ .text: Phần ".text" chứa mã máy (machine code) của chương trình, đây là phần của tệp thực thi chứa các lệnh và mã nhị phân được thực thi bởi máy tính.
+ .data: Phần ".data" chứa dữ liệu tĩnh (static data) của chương trình, bao gồm biến toàn cục và các dữ liệu được khai báo trong mã nguồn.
+ .itext: Phần ".itext" (instruction text) chứa mã máy của chương trình nhưng được chỉ đọc (read-only), nghĩa là nó không thể bị thay đổi trong quá trình thực thi. Thông thường, phần này chứa mã máy không thay đổi, như mã assembly hoặc mã thực thi được mã hóa.
+ .pdata: Phần ".pdata" (procedure data) chứa các thông tin liên quan đến các hàm và quy trình trong chương trình. Nó bao gồm các bản ghi (records) về các hàm, bao gồm địa chỉ bắt đầu và kết thúc của hàm, thông tin liên quan đến xử lý ngoại lệ (exception handling), và các thông tin khác để hỗ trợ quá trình gỡ lỗi và xử lý ngoại lệ.
+ .rsrc: Phần ".rsrc" (resource) chứa các tài nguyên không phải mã máy, chẳng hạn như hình ảnh, biểu đồ, âm thanh, văn bản, biểu mẫu giao diện người dùng, và các tài liệu khác. Đây là nơi chứa các tài nguyên được sử dụng bởi chương trình.

+ .reloc: Phần ".reloc" (relocation) chứa thông tin về việc tương đối (relocation) của mã máy. Khi một tệp thực thi được tải vào bộ nhớ, các địa chỉ tuyệt đối trong mã máy cần được điều chỉnh (relocated) để phù hợp với vị trí cụ thể trong bộ nhớ. Phần ".reloc" chứa các thông tin về việc thay đổi các địa chỉ này khi tệp thực thi được tải vào bộ nhớ.




- Tại đây ta thấy chương trình có sử dụng các thư viện động: ta có thể đưa ra một số giả thuyết liên quan đến việc mã độc có thể thực viễn những hành vi độc hại nào ảnh hưởng đến nạn nhân cũng như cách thức ẩn mình của mã độc

- KERNEL32.DLL: Chứa các hàm tiện ích nhẹ cho giao diện người dùng và các hoạt động liên quan đến chuỗi, đường dẫn, đăng ký và URL.
- USER32.DLL: Chứa các hàm cốt lõi của hệ điều hành Windows, như quản lý bộ nhớ, xử lý, tài nguyên, đồng bộ hóa, xử lý ngoại lệ và nhập xuất. Thư viện này là một phần không thể thiếu của hầu hết các ứng dụng Windows và nếu nó bị hỏng hoặc thiếu, máy tính có thể gặp phải một số lỗi nghiêm trọng.
- SHLWAPI.DLL: Là thư viện chứa các hàm liên quan đến giao diện người dùng của Windows. Như tạo và quản lý cửa sổ, menu, thanh cuộn, nút, hộp thoại bàn phím, chuột và các thông báo. Thư viện này là một phần không thể thiếu của hầu hết các ứng dụng Windows và nếu nó bị hỏng hoặc thiếu, máy tính có thể gặp phải các lỗi khi hiển thị hoặc tương tác với các ứng dụng.

- Đối với việc Zeus Banking Trojan có thể gây ra những hành vi độc hại:

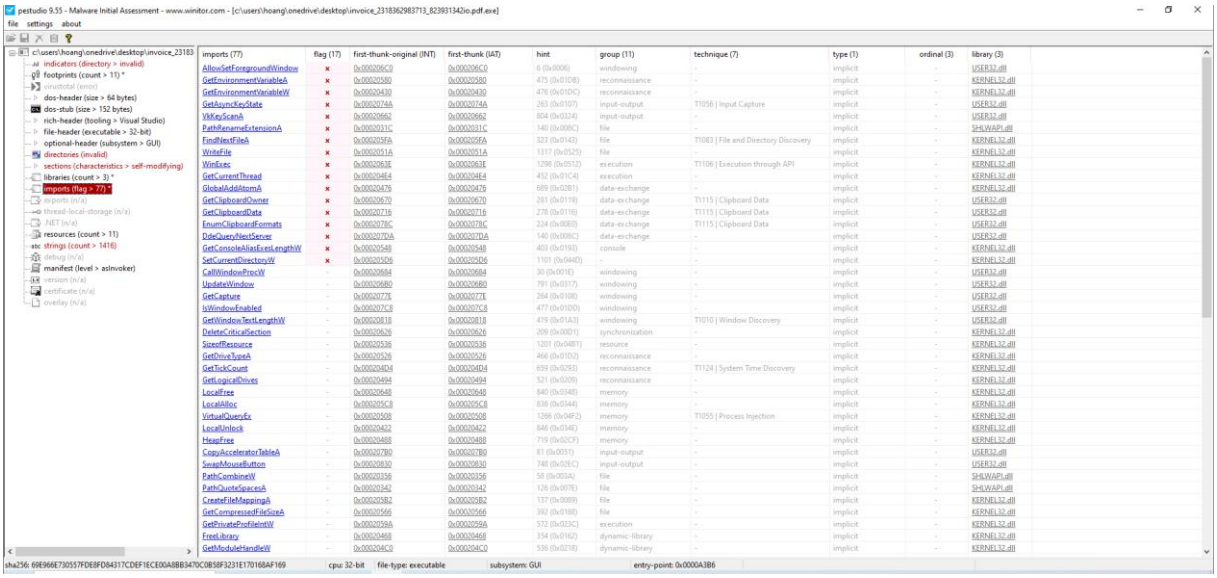
- Zeus banking trojan có thể sử dụng các hàm SHLWAPI.dll để thao tác với các đường dẫn, đăng ký và URL, ví dụ như tạo ra các đường dẫn giả mạo, thay đổi các giá trị đăng ký hoặc kết nối với các máy chủ điều khiển từ xa.

- Zeus banking trojan có thể sử dụng các hàm KERNEL32.dll để thực hiện các hoạt động cốt lõi của hệ điều hành, ví dụ như để cấp phát bộ nhớ, tạo ra các tiến trình con, ghi đè các hàm hệ thống hoặc gọi các hàm API khác.
- Zeus banking trojan có thể sử dụng các hàm USER32.dll để thao tác với giao diện người dùng, ví dụ như theo dõi các sự kiện bàn phím, chuột, cửa sổ, menu, hàn mình hoặc hộp thoại, hoặc để hiển thị các thông báo giả mạo để thu thập các thông tin nhạy cảm từ người dùng.

- Việc Zeus Banking Trojan thực hiện các hành động ẩn mình:

- Zeus banking trojan có thể sử dụng các hàm SHLWAPI.dll để ẩn mình trong các đường dẫn, đăng ký hoặc URL, ví dụ như đổi tên, mã hóa hoặc che dấu các tệp của nó, hoặc để sử dụng các tên mình động hoặc các địa chỉ IP ngẫu nhiên.
- Zeus banking trojan có thể sử dụng các hàm KERNEL32.dll để ẩn mình khỏi các hoạt động cốt lõi của hệ điều hành, ví dụ như đê chèn mã của nó vào các tiến trình hợp lệ, sửa đổi các bảng nhập xuất, hoặc tạo ra các kỹ thuật rootkit.
- Zeus banking trojan có thể sử dụng các hàm USER32.dll để ẩn mình trong giao diện người dùng, ví dụ như để giả lập các ứng dụng, trang web hoặc thông báo hợp lệ hoặc để ẩn các cửa sổ, menu hoặc hộp thoại của nó.

Kiểm tra các imports có thể



Ta có thể chú ý đến một số imports quan trọng: các imports này thường liên quan tới việc truy cập tệp tin, bộ nhớ, thông tin cửa sổ, ghi lại thao tác của người dùng, thực hiện ẩn mình, mở cửa hậu…

- Những hàm mà con mã độc này sử dụng để thực hiện ẩn mình hoặc có các hành vi gây cản trở quá trình phát hiện bởi các công cụ phát hiện mã độc:

- **KERNEL32.SetCurrentDirectoryA** để thay đổi thư mục làm việc hiện tại của tiến trình mã độc sang một thư mục khác, thường là thư mục của một ứng dụng hợp pháp. Điều này giúp mã độc tránh bị phát hiện bởi các công cụ kiểm tra thư mục làm việc của các tiến trình đang chạy.
- **KERNEL32.VirtualQuery** để lấy thông tin về các trang bộ nhớ được sử dụng bởi tiến trình mã độc, sau đó sử dụng hàm **KERNEL32.ConvertDefaultLocale** để mã hóa các trang bộ nhớ đó bằng một thuật toán XOR. Điều này giúp mã độc tránh bị phát hiện bởi các công cụ kiểm tra bộ nhớ của các tiến trình đang chạy.
- **KERNEL32.CreateIoCompletionPort** để tạo một cổng hoàn thành I/O, sau đó sử dụng hàm **KERNEL32.OpenFileMappingA** để mở một ánh xạ tệp được tạo bởi một tiến trình khác. Điều này giúp mã độc tránh bị phát hiện bởi các công cụ kiểm tra các tệp được mở bởi các tiến trình đang chạy.
- **KERNEL32.DeleteFileA** để xóa tệp thực thi của mã độc sau khi nó được chạy. Điều này giúp mã độc tránh bị phát hiện bởi các công cụ kiểm tra các tệp có nghi ngờ trên đĩa cứng.

- Những hàm mà con mã độc này sử dụng để thực hiện các hành vi độc hại đối với nạn nhân:

- **SHLWAPI.PathMakeSystemFolderW** để tạo một thư mục hệ thống ẩn, sau đó sử dụng hàm **SHLWAPI.PathAddExtensionA** để thêm một phần mở rộng tệp giả vào tên của thư mục đó. Điều này giúp mã độc lừa nạn nhân vào việc mở thư mục đó như một tệp thực thi, từ đó kích hoạt mã độc.
- **USER32.GetShellWindow** để lấy cửa sổ của shell Windows, sau đó sử dụng hàm **USER32.GetPropW** để lấy các thuộc tính của cửa sổ đó. Điều này giúp mã độc lấy được các thông tin về phiên làm việc của nạn nhân, như tên người dùng, tên máy tính, phiên bản hệ điều hành, v.v.
- **USER32.SetDlgItemTextW** để thay đổi nội dung của các hộp thoại trên các trang web của các ngân hàng hoặc các dịch vụ thanh toán trực tuyến. Điều này giúp mã độc lấy được các thông tin nhạy cảm của nạn nhân, như số tài khoản, mật khẩu, mã xác thực, v.v.
- **USER32.GetMonitorInfoW** để lấy thông tin về kích thước và độ phân giải của màn hình, sau đó sử dụng hàm **USER32.GetUpdateRgn** để lấy vùng cập nhật của một cửa sổ. Điều này giúp mã độc chụp ảnh màn hình của nạn nhân, từ đó thu thập thêm các thông tin về hoạt động của nạn nhân trên internet.

- Một số hàm có thể thể hiện cho việc ZeuS banking mở cửa hậu hoặc gửi dữ liệu thu thập được trên máy nạn nhân về máy tính của kẻ tấn công là:

- **KERNEL32.OpenFileMappingA**: Hàm này cho phép mã độc mở một ánh xạ tệp được tạo bởi một tiến trình khác, có thể là một cửa hậu đã được cài đặt trước đó. Hàm này giúp mã độc truy cập vào các tệp hoặc bộ nhớ được chia sẻ bởi cửa hậu, từ đó có thể gửi dữ liệu ra ngoài hoặc nhận lệnh từ kẻ tấn công.
- **SHLWAPI.SHLockShared** và **SHLWAPI.SHFreeShared**: Hai hàm này cho phép mã độc khóa và giải phóng một đối tượng được chia sẻ bởi nhiều tiến trình. Hàm này giúp mã độc đồng bộ hóa việc gửi và nhận dữ liệu qua cửa hậu, tránh xung đột hoặc mất mát dữ liệu.
- **USER32.GetShellWindow**: Hàm này cho phép mã độc lấy cửa sổ của shell Windows, là một cửa sổ ẩn chứa các icon và thanh tác vụ. Hàm này giúp mã độc tạo ra một cửa sổ con bên trong cửa sổ shell, từ đó có thể gửi và nhận dữ liệu qua mạng mà không bị phát hiện bởi người dùng hoặc các công cụ bảo mật.

- Bên cạnh đó còn có thể kể đến 1 số những hàm khác:

- KERNEL32.PathRelativePathToW: dùng để tạo một đường dẫn tương đối từ một tệp hoặc đường thư mục khác, mã độc có thể sử dụng hàm này để xác định vị trí của các tệp hoặc thư mục quan trọng trên hệ thống nạn nhân như tệp cấu hình, tệp nhật ký, hoặc tệp chứa thông tin tài khoản ngân hàng, sau đó mã độc có thể đọc, ghi, hoặc xóa các tệp hoặc thư mục này để thực hiện các hành vi độc hại.
- KERNEL32.OpenFileMappingA: dùng để mở một đối tượng ánh xạ tệp haojcw không có tên cho 1 tệp chỉ định, Zeus Banking Trojan co thể sử dụng hàm này để tạo 1 vùng nhớ chia sẽ giữa các tiến trình, hoặc giữa các máy tính thông qua mạng, điều này giúp mã độc truyền dữ liệu vè cho kẻ tấn công hoặc lây nhiễm sang các máy tính khác, ngoài ra hàm này cũng có thể giúp mã độc ẩn mình, bằng cách tạo 1 đối tượng ánh xạ tệ không có tên, và sử dụng các hàm như MapViewOfFile để truy cập vào vùng nhớ đó, khiến cho các công cụ bảo mật khó lòng phát hiện.

- Kiểm tra exports:

| file-names | |
|---|---|
| export | corect.com |
| debug | n/a |
| version | n/a |
| manifest | n/a |
| .NET | n/a |

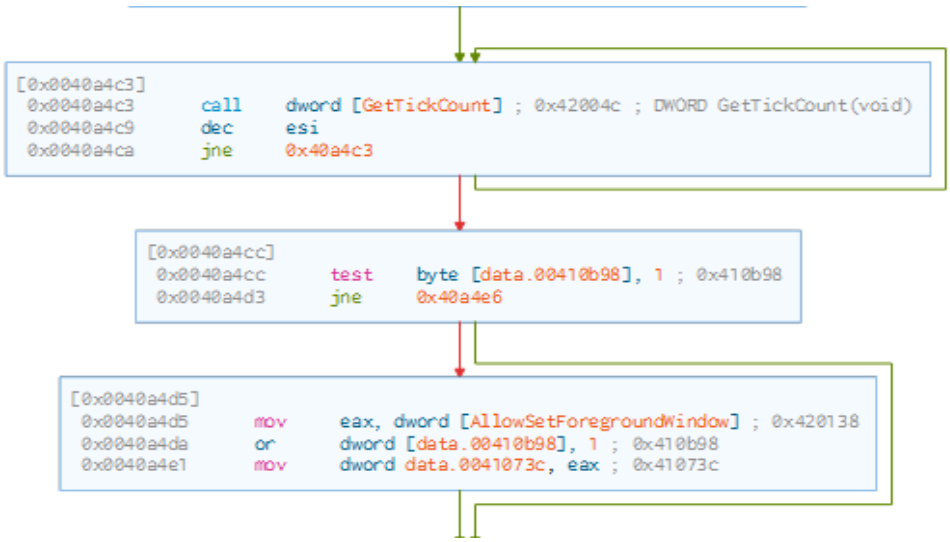
Có thể đây là nơi dữ liệu bị đánh cắp gửi đến.

```
C:\Users\hoang\OneDrive\Desktop
λ capa invoice_2318362983713_823931342io.pdf.exe
┌──────────┬──────────────────────────────────────────────────────────────────────────────────────────────────────┐
│ md5      │ ea039a854d20d7734c5add48f1a51c34                                                                     │
│ sha1     │ 9615dca4c0e46b8a39de5428af7db060399230b2                                                             │
│ sha256   │ 69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169                                     │
│ os       │ windows                                                                                              │
│ format   │ pe                                                                                                   │
│ arch     │ i386                                                                                                 │
│ path     │ C:/Users/hoang/OneDrive/Desktop/invoice_2318362983713_823931342io.pdf.exe                            │
└──────────┴──────────────────────────────────────────────────────────────────────────────────────────────────────┘

┌──────────────┬──────────────────────────────────────────────────────────────┐
│ ATT&CK Tactic │ ATT&CK Technique                                            │
├──────────────┼──────────────────────────────────────────────────────────────┤
│ DEFENSE EVASION │ Virtualization/Sandbox Evasion::System Checks T1497.001   │
└──────────────┴──────────────────────────────────────────────────────────────┘

┌──────────────┬──────────────────────────────────────────────────────────────┐
│ MBC Objective │ MBC Behavior                                                │
├──────────────┼──────────────────────────────────────────────────────────────┤
│ ANTI-BEHAVIORAL ANALYSIS │ Virtual Machine Detection [B0009]                │
└──────────────┴──────────────────────────────────────────────────────────────┘

┌──────────────────────────────────────────┬──────────────────────────────────┐
│ Capability                               │ Namespace                        │
├──────────────────────────────────────────┼──────────────────────────────────┤
│ reference anti-VM strings targeting VMWare │ anti-analysis/anti-vm/vm-detection │
│ resolve function by parsing PE exports   │ load-code/pe                     │
└──────────────────────────────────────────┴──────────────────────────────────┘
```

- Sử dụng Capa để tìm phương thức phòng thủ thì ta thấy nó đã sử dụng kỹ thuật sử dụng kỹ thuật VM/Sandbox Evasion để tránh bị kiểm tra và phát hiện.
- Ta còn thấy rằng mã độc này còn có khả năng phát hiện môi trường ảo : Virtual Machine Detection.
- Sử dụng Cutter sẽ hiển thị cho chúng ta các đoạn mã , giúp ta biết được mã độc này đang muốn làm gì.

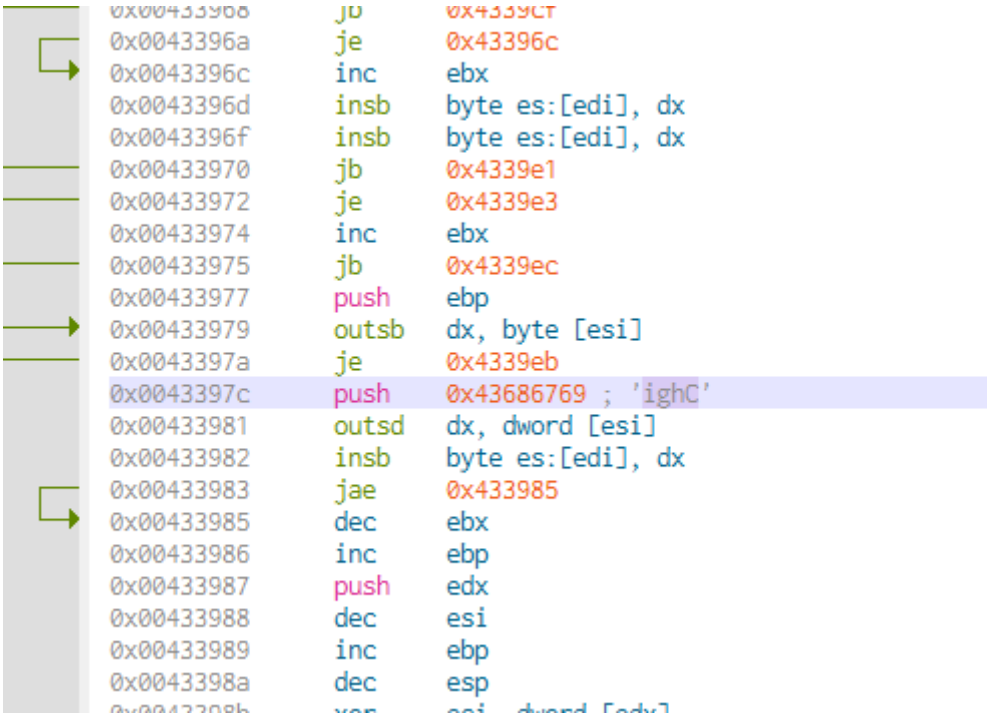
```
0x0040a4be      mov     esi, 0x80000
0x0040a4c3      call    dword [GetTickCount] ; 0x42004c ; DWORD GetTickCount(void)
0x0040a4c9      dec     esi
0x0040a4ca      jne     0x40a4c3
0x0040a4cc      test    byte [data.00410b98], 1 ; 0x410b98
0x0040a4d3      jne     0x40a4e6
0x0040a4d5      mov     eax, dword [AllowSetForegroundWindow] ; 0x420138
0x0040a4da      or      dword [data.00410b98], 1 ; 0x410b98
0x0040a4e1      mov     dword data.0041073c, eax ; 0x41073c
0x0040a4e6      mov     dword [var_94h], 0x5f31 ; '1_'
```

- Hàm GetTickCount() kiểm tra xem máy Windows đã chạy được bao lâu. Đây có thể là hàm giúp cho mã độc phát hiện được ra môi trường có phải là môi trường ảo hay không và lẩn tránh. Đây cũng có thể là để mã độc vào chế độ ngủ đông, tránh gây việc theo dõi trong quá trình phân tích động. Sau đó một thời gian sẽ tự khởi động chạy hàm AllowSetForegroundWindow cho mã độc quyền chạy mà không bị giới hạn bởi các quy tắc an toàn trong hệ điều hành, vượt qua bảo mật hệ thống.

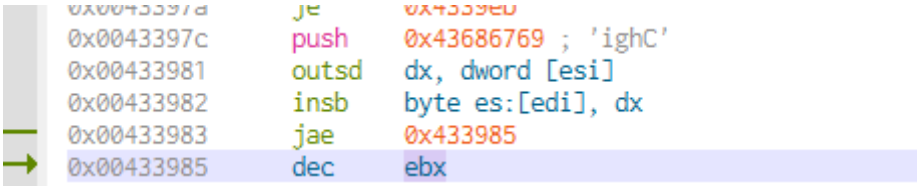



- Nhìn vào các chuỗi ngẫu nhiên được trích xuất :

CellrotoCrudUntohighCols

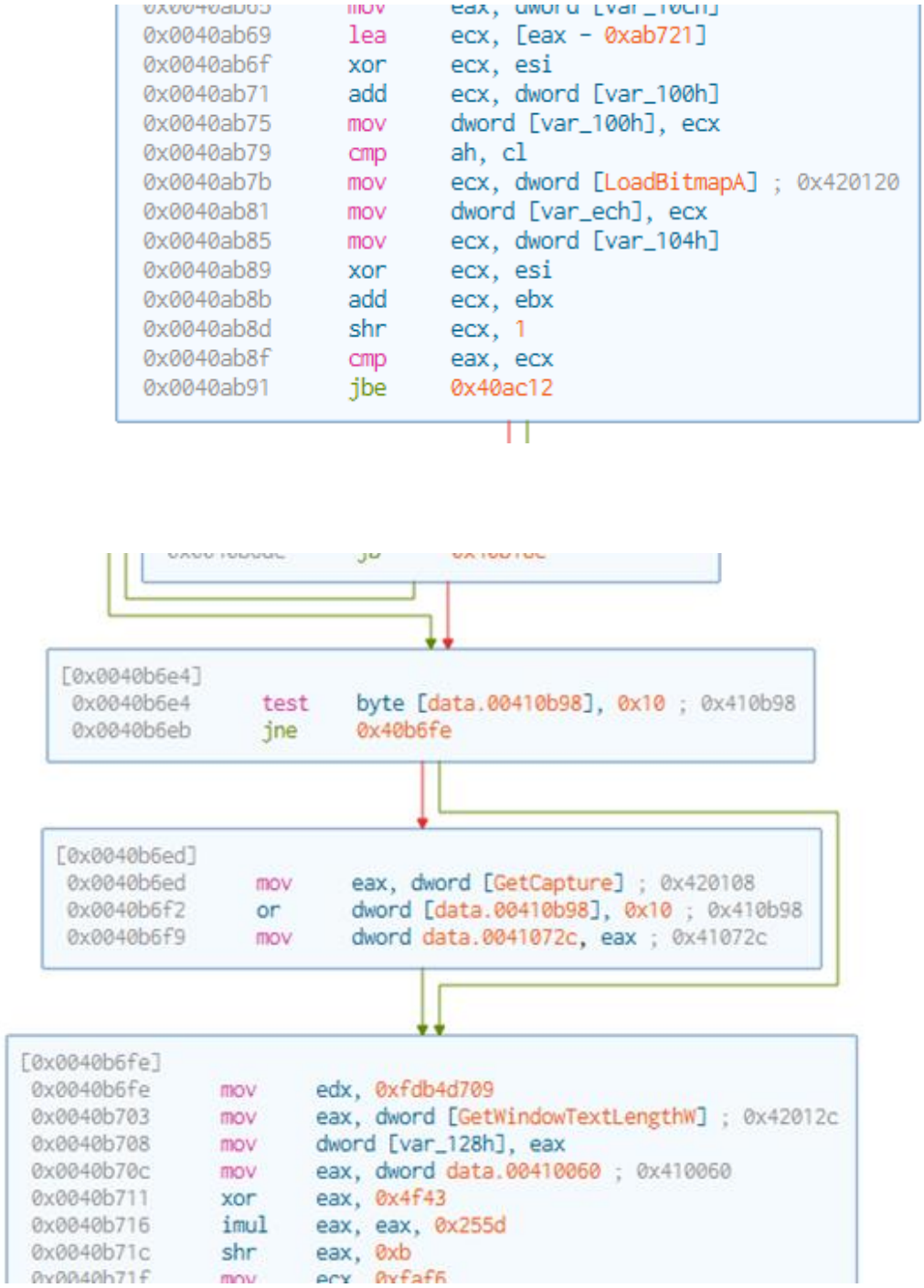



KERNEL32.CreateFileA

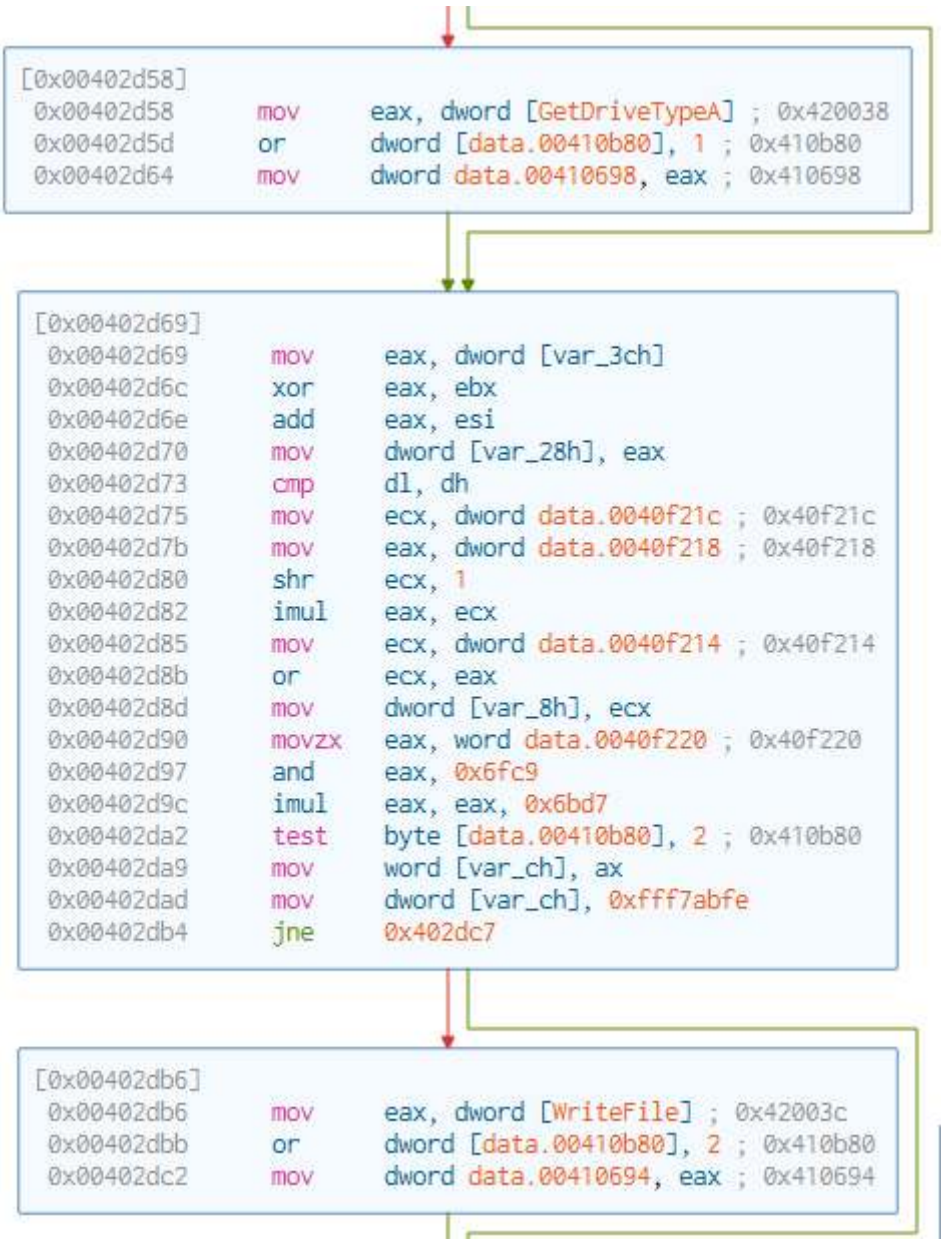



Ta sẽ thử tìm trong cutter xem dòng rác này có gì đặc biệt không và khi xem ở mục disassemble t thấy nó có đoạn mã string ‘ighC’ được sử dụng nhằm mục đích gì đó và ngay gần dưới đó là hàm KERNEL32.CreateFileA nên ta có giả thuyết ở đây là mã độc đang cố tạo hoặc mở một thư mục gì đó nhằm các mục đích gây hại. Đây mới chỉ

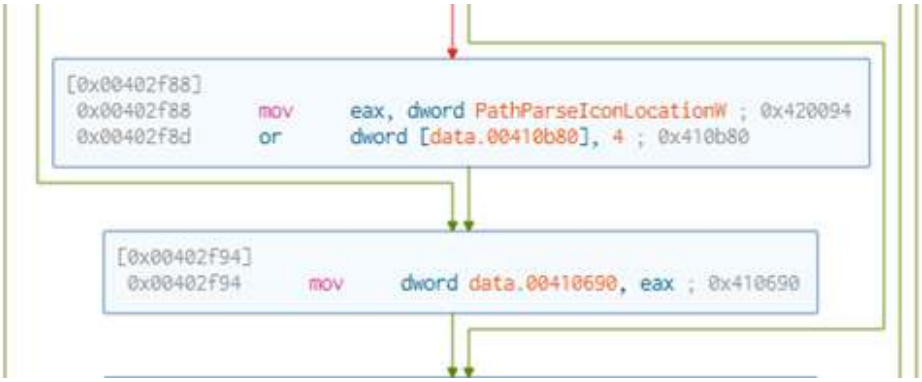
là giả thuyết và còn rất nhiều dòng lệnh ngẫu nhiên như thế đi kèm theo các hàm gọi có nghĩa.




LoadBitmap và GetCapture có thể sử dụng để lấy cắp hình ảnh trên màn hình máy nhiễm độc, có khả năng để xem thông tin cá nhân hoặc thông tin đăng nhập của người dùng



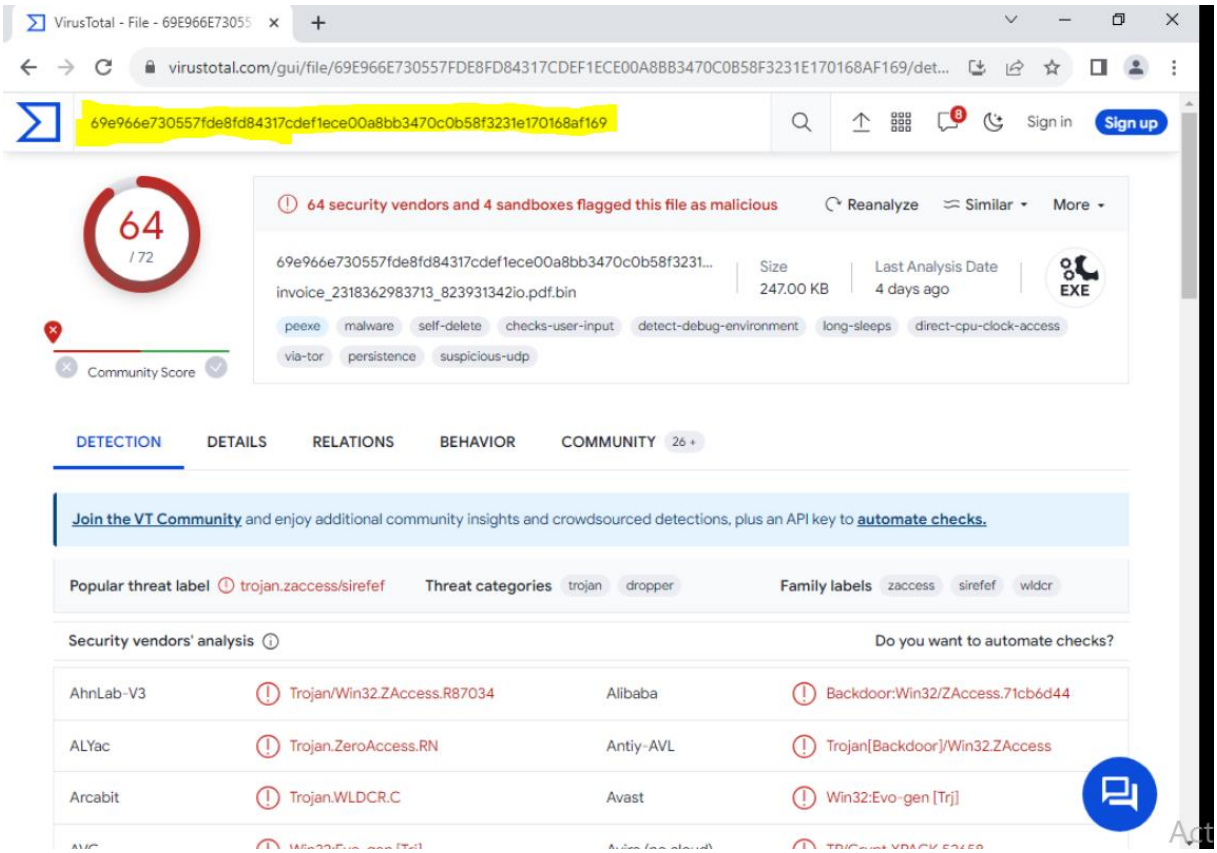

GetDriveType có thể dùng để phân biệt loại của drive trên máy tính bị nhiễm mã độc, khả năng nếu đấy là USB drive thì mã độc sẽ thực thi lnehj WriteFile để lây nhiễm qua USB rồi qua các máy khác qua con USB này



PathParseIconLationW được dùng để phân tích vị trí file cũng như có thể dùng để lấy icon file để giả mạo một chương trình tin cậy khác trên hệ thống.

8. Phân loại

Phân loại dựa trên chữ ký: dựa vào virustotal, so sánh mã Hash hoặc các đoạn mã đặc trưng của tập tin nghi ngờ với cơ sở dữ liệu chứa chữ ký của các loại mã độc đã biết.




=> nếu có thể quét được, ta có thể biết được mã độc này thuộc loại nào, họ nào, thực hiện các hành vi độc hại nào trên máy tính của nạn nhân

Những mã độc thực hiện các hoạt động đánh cắp thông tin tài khoản ngân hàng, Zeus banking trojan có thể theo dõi, hị lại các hoạt động của người dùng trên các trang web ngân hàng, và gửi thông tin đó về cho kẻ tấn công. Ngoài ra nso cũng có thể thay đổi nội dung của các trang web ngân hàng để lừa người dùng nhập các thông tin nhạy cảm, như mã OTP, mã PIN

| | | | |
|---|---|---|---|
| Microsoft | TrojanDropper:Win32/Sirefef.gen!B | NANO-Antivirus | Trojan.Win32.ZAccess.cqjtnp |
| Panda | Trj/WLT.A | QuickHeal | TrojanDropper.Sirefef.A9 |
| Rising | Dropper.Sirefef!8.525 (TFE:2:1ubYWh7DM... | Sangfor Engine Zero | Suspicious.Win32.Save.a |

Zeus banking trojan cũng là 1 loại mã độc mở cửa hậu, nó có thể tạo ra một kết nối mạng giữa máy tính nạn nhân và máy tính của kẻ tấn công, cho phép kẻ tấn công điều khiển máy tính của nạn nhân từ xa, nó cũng có thể tải xuống và chạy các tệp độc hại khác từ máy tính của kẻ tấn công

| | |
|---|---|
| Kaspersky | Backdoor.Win32.ZAccess.evyo |

| | |
|---|---|
| SUPERAntiSpyware | Backdoor.ZAccess/Variant |

| | | | |
|---|---|---|---|
| Yandex | Backdoor.ZAccess!UkzQ0/sevQU | Zillya | Backdoor.ZAccess.Win32.30281 |
| ZoneAlarm by Check Point | Backdoor.Win32.ZAccess.evyo | Zoner | Trojan.Win32.19899 |

| | | | |
|---|---|---|---|
| TrendMicro | BKDR_SIREFEF.NIS | TrendMicro-HouseCall | BKDR_SIREFEF.NIS |

Thấy hành vi ẩn mình gây cản trở quá trình phát hiện của con mã độc này:

Zeus banking trojan thực hiện các kỹ thuật như rootkit, polymorphism, obfusaction …, nó co thể tự xóa hoặc tự sao chép để trốn tránh các công cụ bảo mật, ngoài ra một số cái tên khác của mã độc này có thể kể đến:

| | | | |
|---|---|---|---|
| K7AntiVirus | RootKit ( 004b9ee21 ) | K7GW | RootKit ( 004b9ee21 ) |

| | | | |
|---|---|---|---|
| Arcabit | Trojan.WLDCR.C | Avast | Win32:Evo-gen [Trj] |
| AVG | Win32:Evo-gen [Trj] | Avira (no cloud) | TR/Crypt.XPACK.52658 |
| BitDefender | Trojan.WLDCR.C | BitDefenderTheta | Gen:NN.ZexaF.36792.pyW@aqPTyGbO |
| Bkav Pro | W32.AIDetectMalware | CrowdStrike Falcon | Win/malicious_confidence_100% (W) |
| Cybereason | Malicious.4c0e46 | Cylance | Unsafe |
| Cynet | Malicious (score: 99) | DeepInstinct | MALICIOUS |
| DrWeb | BackDoor.Maxplus.14813 | Elastic | Malicious (high Confidence) |
| Emsisoft | Trojan.WLDCR.C (B) | eScan | Trojan.WLDCR.C |

| | | | |
|---|---|---|---|
| VIPRE | Trojan.WLDCR.C | VirIT | Trojan.Win32.Dropper.JH |

| | | | |
|---|---|---|---|
| Arcabit | Trojan.WLDCR.C | Avast | Win32:Evo-gen [Trj] |
| AVG | Win32:Evo-gen [Trj] | Avira (no cloud) | TR/Crypt.XPACK.52658 |

| | | | |
|---|---|---|---|
| ALYac | Trojan.ZeroAccess.RN | Antiy-AVL | Trojan[Backdoor]/Win32.ZAccess |

Những mã độc thự hiện gửi thông tin về máy tính của tin tặc: Zeus banking có thể gửi thông tin về máy tính của kẻ tấn công thông qua các kênh khác nhau như HTTP, FTP, SMTP …, nó có thể sử dụng các kỹ thuật mã hóa, nén hoặc ẩn thông tin trong các gói tin mạng để tránh bị phát hiện, tên gọi khác của mã độc này có thể kể đến:

Mối đe dọa phổ biến

| | |
|---|---|
| ESET-NOD32 | Win32/Sirefef.FY |

| | |
|---|---|
| Avira (no cloud) | TR/Crypt.XPACK.52658 |
| BitDefenderTheta | Gen:NN.ZexaF.36792.pyW@aqPTyGbO |
| CrowdStrike Falcon | Win/malicious_confidence_100% (W) |
| Cylance | Unsafe |
| DeepInstinct | MALICIOUS |
| Elastic | Malicious (high Confidence) |
| eScan | Trojan.WLDCR.C |
| F-Secure | Trojan.TR/Crypt.XPACK.52658 |

| | |
|---|---|
| Fortinet | W32/Generic.AC.3F8DF6!tr |

# V. TỔNG KẾT

| Nguyên tắc hoạt động / Dấu hiệu hành vi. | Xâm nhập vào hệ thống | Điều khiển và kiểm soát hệ thống | Mở cửa hậu cho kẻ tấn công | Thực hiện các hoạt động độc hại | Che giấu sự hoạt động |
|---|---|---|---|---|---|
| Giả dạng file pdf | x | | | | x |
| Đóng gói, Cố gắng che giấu các chuỗi | | | | | x |
| Sử dụng các thư viện động | | | | x | x |
| Sử dụng các hàm đặc quyền | x | x | x | x | x |
| Sử dụng Unti VM/Sandbo | | | | | x |
| | | | | | |
| | | | | | |
| | | | | | |