BÁO CÁO BÀI TẬP 2

**Môn học: An toàn mạng máy tính nâng cao**

**BT2:**

**Tiếp cận cơ chế hoạt động của mã độc và sử dụng công cụ để phân tích hoạt động của mã độc.**

**Botnet + capa**

*GVHD: Nguyễn Duy*

1. **THÔNG TIN CHUNG:**

Lớp: NT534.N21.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hoàng Văn Anh Đức | 20520890 | 20520890@gm.uit.edu.vn |
| 2 | Nguyễn Tú Ngọc | 20521665 | 20521665@gm.uit.edu.vn |
| 3 | Trần Đặng Hồng Loan | 20521543 | 20521543@gm.uit.edu.vn |
| 4 | Lê Nhật Hào | 20520489 | 20520489@gm.uit.edu.vn |

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

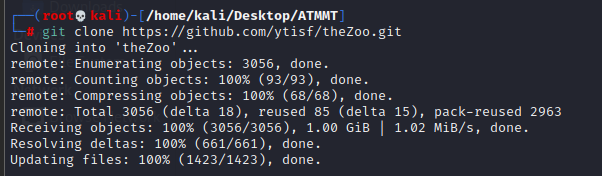
|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Download và sử dụng Kho Malware: theZoo: A live malware repository để tải malware | 100% |
| 2 | Download “capa: Automatically identify malware capabilities” để phân tích malware | 100% |
| 3 | Sử dụng capa đánh giá và nhận xét kết quả thu được từ files | 100% |

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

BÁO CÁO CHI TIẾT

**1. Download và sử dụng Kho Malware: theZoo: A live malware repository để tải malware**

Tải kho malware trên github : <https://github.com/ytisf/theZoo.git>



Vào thư mục để tải các requirements

Text

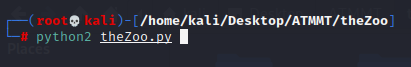
Description automatically generated

Đã cài xong

A picture containing graphical user interface

Description automatically generated

Sử dụng theZoo.py để tạo malware



Đọc các cảnh báo trước khi sử dụng

Text

Description automatically generated

Search botnet

Text

Description automatically generated

List các botnet có thể tải về dùng

A picture containing graphical user interface

Description automatically generated

Chọn botnet và tải về

Text

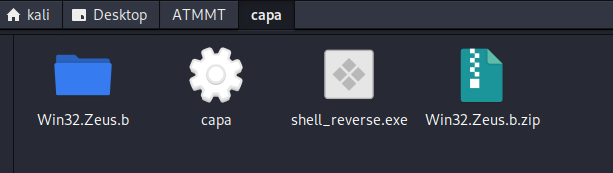
Description automatically generated

**2.Download và sử dụng “capa: Automatically identify malware capabilities” để phân tích malware**

<https://github.com/mandiant/capa/releases>

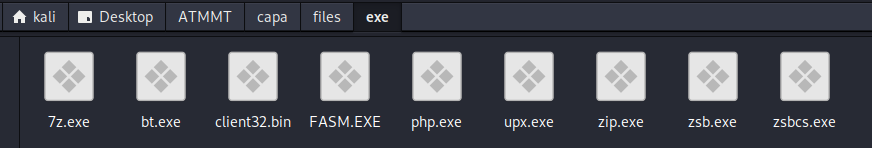


Tải về và unzip

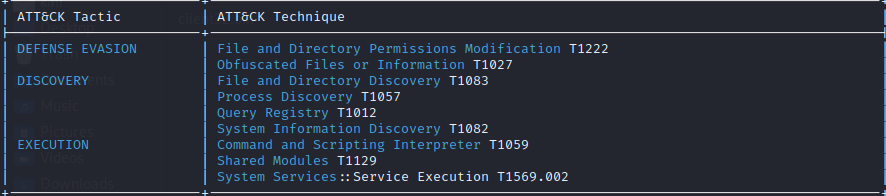


* Tải thành công capa

**3.Sử dụng capa đánh giá và nhận xét kết quả thu được từ files**

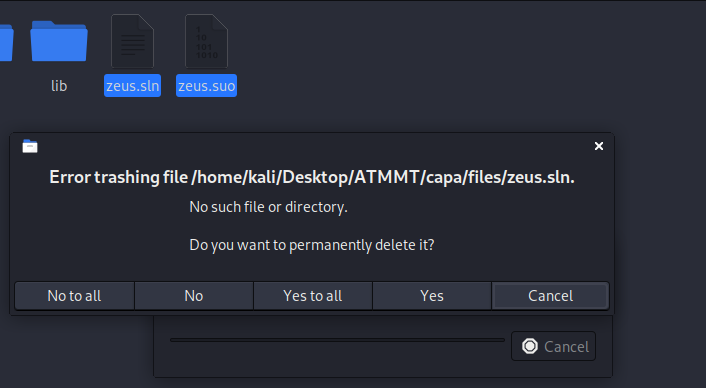


7z.exe



* "**DEFENSE EVASION**" : giai đoạn mà tấn công được thực hiện để đánh lừa hoặc tránh các biện pháp phòng ngự của hệ thống
  + **File and Directory Permissions Modification T1222**: Đây là một kỹ thuật tấn công sử dụng để thay đổi quyền truy cập của các tệp tin và thư mục trên hệ thống. Kẻ tấn công có thể sử dụng kỹ thuật này để cài đặt mã độc hoặc ngăn chặn người dùng truy cập vào các tệp tin hoặc thư mục quan trọng.
  + **Obfuscated Files or Information T1027**: sử dụng để ẩn giấu mã độc hoặc thông tin bằng cách thay đổi cấu trúc hoặc mã hóa dữ liệu. Kẻ tấn công có thể sử dụng kỹ thuật này để tránh bị phát hiện bởi các giải pháp bảo mật.

Ví dụ : Không xóa đc các file của chương trình virus khi nó có trong máy



* "**DISCOVERY**" là giai đoạn trong đó hacker sẽ khám phá hệ thống của mục tiêu để tìm kiếm các thông tin hữu ích hoặc các lỗ hổng bảo mật
  + **File and Directory Discovery T1083**: sử dụng để tìm kiếm các tệp tin và thư mục trên hệ thống. Kẻ tấn công có thể sử dụng kỹ thuật này để tìm các tệp tin quan trọng hoặc các dữ liệu mật.
  + **Process Discovery T1057**: để tìm kiếm các tiến trình đang chạy trên hệ thống. Kẻ tấn công có thể sử dụng kỹ thuật này để tìm các tiến trình quan trọng hoặc để tìm cách thực thi mã độc.
  + **Query Registry T1012**: sử dụng để truy vấn các giá trị trong registry của hệ thống. Kẻ tấn công có thể sử dụng kỹ thuật này để tìm kiếm các thông tin quan trọng hoặc để tìm cách thực thi mã độc.
  + **System Information Discovery T1082**: sử dụng để tìm kiếm các thông tin hệ thống quan trọng, chẳng hạn như phiên bản hệ điều hành, kiến trúc hệ thống, các ứng dụng đang chạy, v.v. Kẻ tấn công có thể sử dụng kỹ thuật này để tìm kiếm các thông
* "**EXECUTION**" là giai đoạn cuối cùng trong đó hacker thực hiện các hành động nhằm đạt được mục tiêu của mình
  + **Command and Scripting Interpreter T1059**: Kỹ thuật này tấn công vào khả năng thực thi các lệnh và script trên hệ thống bằng cách sử dụng các trình thông dịch như cmd.exe, PowerShell, Bash, Perl, Python, v.v. Nó cho phép kẻ tấn công thực thi các lệnh độc hại hoặc thu thập thông tin từ hệ thống mục tiêu. Kỹ thuật này thường được sử dụng trong các cuộc tấn công phức tạp như các cuộc tấn công mạng tiên tiến (APT).
  + **Shared Modules T1129**: Kỹ thuật này tấn công vào khả năng sử dụng các thư viện chia sẻ bằng cách chèn mã độc vào các tập tin thư viện chia sẻ. Khi ứng dụng sử dụng các thư viện này, mã độc sẽ được thực thi, cho phép kẻ tấn công kiểm soát máy tính mục tiêu.
  + **System Services::Service Execution T1569.002:** Kỹ thuật này tấn công vào khả năng thực thi các dịch vụ hệ thống bằng cách sử dụng các ứng dụng hệ thống hoặc các công cụ quản lý dịch vụ để chạy các dịch vụ với quyền hạn cao hơn. Nó cho phép kẻ tấn công thực hiện các hoạt động độc hại trên hệ thống mục tiêu, chẳng hạn như cài đặt phần mềm độc hại hoặc thu thập thông tin.

Text

Description automatically generated

* **DATA**: sử dụng kỹ thuật mã hóa XOR để mã hóa dữ liệu.
* **DEFENSE EVASION**: sử dụng kỹ thuật mã hóa để ẩn thông tin hay mã độc trên hệ thống. Trong trường hợp này, kỹ thuật mã hóa được sử dụng là Encoding-Standard Algorithm (E1027.m02).
* **DISCOVERY**: sử dụng kỹ thuật khám phá tập tin và thư mục trên hệ thống (File and Directory Discovery [E1083]) để thu thập thông tin về các tài liệu, chương trình và cấu hình của hệ thống mục tiêu.
* **EXECUTION**: sử dụng kỹ thuật thực thi mã lệnh và script (Command and Scripting Interpreter [E1059]) để thực hiện các lệnh và kịch bản độc hại trên hệ thống mục tiêu. FILE SYSTEM: MBC sử dụng nhiều hoạt động liên quan đến hệ thống tập tin như tạo thư mục, xóa thư mục, xóa tập tin, di chuyển tập tin, đọc tập tin, thiết lập thuộc tính tập tin, viết tập tin.
* **OPERATING SYSTEM**: sử dụng kỹ thuật truy vấn giá trị Registry (Registry::Query Registry Value [C0036.006]) để truy cập thông tin trong Registry trên hệ thống mục tiêu.
* **PROCESS**: sử dụng kỹ thuật tạo tiểu trình (Create Thread [C0038]) để thực thi các tiến trình độc hại trên hệ thống mục tiêu.

A screenshot of a computer

Description automatically generated with medium confidence

**encode data using XOR**: đối tượng có khả năng mã hóa dữ liệu bằng XOR, nằm trong namespace data-manipulation/encoding/xor.

**contain a resource (.rsrc) section**: đối tượng chứa một phần tài nguyên (.rsrc), nằm trong namespace executable/pe/section/rsrc.

**accept command line arguments**: đối tượng có khả năng chấp nhận tham số dòng lệnh, nằm trong namespace host-interaction/cli.

**interact with driver via control codes**: đối tượng có khả năng tương tác với trình điều khiển thông qua các mã điều khiển, nằm trong namespace host-interaction/driver.

**get common file path (9 matches):** đối tượng có khả năng lấy đường dẫn tệp tin phổ biến, có 9 kết quả khớp với namespace host-interaction/file-system.

**set current directory**: đối tượng có khả năng thiết lập thư mục hiện tại, nằm trong namespace host-interaction/file-system.

**create directory (2 matches):** đối tượng có khả năng tạo thư mục, có 2 kết quả khớp với namespace host-interaction/file-system/create.

**delete directory**: đối tượng có khả năng xóa thư mục, nằm trong namespace host-interaction/file-system/delete.

**delete file (2 matches):** đối tượng có khả năng xóa tệp tin, có 2 kết quả khớp với namespace host-interaction/file-system/delete. get file size: đối tượng có khả năng lấy kích thước tệp tin, nằm trong namespace host-interaction/file-system/meta.

**set file attributes (3 matches):** đối tượng có khả năng thiết lập thuộc tính tệp tin, có 3 kết quả khớp với namespace host-interaction/file-system/meta.

**move file (2 matches):** đối tượng có khả năng di chuyển tệp tin, có 2 kết quả khớp với namespace host-interaction/file-system/move.

**read file on Windows (2 matches):** đối tượng có khả năng đọc tệp tin trên hệ điều hành Windows, có 2 kết quả khớp với namespace host-interaction/file-system/read.

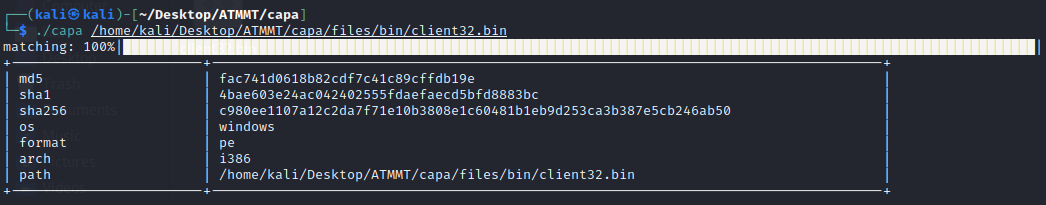
**read file via mapping**: đối tượng có khả năng đọc tệp tin thông qua bản đồ, nằm trong namespace host-interaction/file-system/read.

**write file on Windows (2 matches)**: đối tượng có khả năng ghi tệp tin trên hệ điều hành Windows, có 2 kết quả khớp với namespace host-interaction/file-system/write.

**get memory capacity**: đối tượng có khả năng lấy dung lượng bộ nhớ, nằm trong namespace

Graphical user interface, text

Description automatically generated



Text

Description automatically generated

* **COLLECTION**: Chứa hai kỹ thuật sử dụng để thu thập thông tin từ hệ thống đích.
  + **Clipboard Data** (T1115), được sử dụng để thu thập dữ liệu được sao chép vào bộ nhớ đệm (clipboard) trên hệ thống đích.
  + **Input Capture**::Keylogging (T1056.001), được sử dụng để ghi lại các phím được ấn trên bàn phím của hệ thống đích.
* **DEFENSE EVASION**: Chứa bốn kỹ thuật được sử dụng để ngăn chặn hoặc tránh các biện pháp phòng ngự của hệ thống đích.
  + **File and Directory Permissions Modification (T1222),** được sử dụng để thay đổi quyền truy cập của tệp và thư mục trên hệ thống đích.
  + **Modify Registry (T1112),** được sử dụng để thay đổi các giá trị trong registry trên hệ thống đích.
  + **Obfuscated Files or Information (T1027),** được sử dụng để giấu thông tin hoặc mã độc trên hệ thống đích.
  + **Obfuscated Files or Information::Indicator Removal from Tools (T1027.005)**, được sử dụng để xóa các tín hiệu (indicators) của công cụ tấn công khỏi các tệp trên hệ thống đích.
* **DISCOVERY**: Chứa 9 kỹ thuật được sử dụng để khám phá thông tin về hệ thống đích. Các kỹ thuật này bao gồm:
  + **Account Discovery (T1087),** được sử dụng để tìm kiếm các tài khoản trên hệ thống đích;
  + **File and Directory Discovery (T1083),** được sử dụng để tìm kiếm các tệp và thư mục trên hệ thống đích;
  + **Process Discovery (T1057)**, được sử dụng để tìm kiếm các quá trình đang chạy trên hệ thống đích;
  + **Query Registry (T1012),** được sử dụng để truy vấn registry trên hệ thống đích
  + **Software Discovery T1518**: kỹ thuật này dùng để tìm kiếm và thu thập thông tin về phần mềm được cài đặt trên hệ thống đích. System
  + **Information Discovery T1082**: kỹ thuật này dùng để thu thập thông tin về hệ thống đích như tên máy tính, hệ điều hành, phiên bản, thông tin phần cứng, ...
  + **System Location Discovery::System Language Discovery T1614.001**: kỹ thuật này dùng để xác định vị trí địa lý của hệ thống đích và ngôn ngữ được sử dụng.
  + **System Network Configuration Discovery T1016**: kỹ thuật này dùng để tìm kiếm và thu thập thông tin về cấu hình mạng của hệ thống đích.
  + **System Owner/User Discovery T1033**: kỹ thuật này dùng để tìm kiếm và thu thập thông tin về các người dùng và nhóm người dùng trên hệ thống đích.

A screenshot of a computer

Description automatically generated with medium confidence

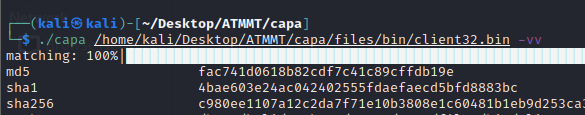
* **ANTI-BEHAVIORAL ANALYSIS:**
  + "**Debugger Detection::Timing/Delay Check GetTickCount [B0001.032**]" là một phương pháp để phát hiện xem có (debugger) nào đang chạy trên hệ thống hay không. Phương pháp này sử dụng hàm GetTickCount() để tính thời gian giữa hai lần gọi hàm liên tiếp. Nếu thời gian này ngắn hơn một ngưỡng nhất định, điều đó có thể cho thấy rằng một trình gỡ lỗi đang chạy trên hệ thống, vì hầu hết các trình gỡ lỗi sẽ làm chậm thời gian giữa hai lần gọi hàm.
* **COLLECTION**
  + **| Keylogging::Polling [F0002.002]** đây là một hoạt động giám sát và thu thập dữ liệu về các phím được nhấn trên bàn phím của người
* **COMMAND AND CONTROL**
  + **C2 Communication::Receive Data [B0030.002**] : các hoạt động liên lạc điều khiển và điều chỉnh của phần mềm độc hại.
    - Phần mềm độc hại có khả năng gửi và nhận dữ liệu với máy chủ điều khiển và điều chỉnh thông qua kênh truyền C2.
    - B0030.002 đề cập đến khả năng nhận dữ liệu từ máy chủ C2
  + **C2 Communication::Send Data [B0030.001**] : khả năng gửi dữ liệu đến máy chủ C2.
* **COMMUNICATION** 
  + DNS Communication::Resolve [C0011.001]: phát hiện và giải quyết tên miền DNS.
  + HTTP Communication::Connect to Server [C0002.009]: kết nối đến máy chủ HTTP.
  + HTTP Communication::Create Request [C0002.012]: tạo yêu cầu HTTP.
  + HTTP Communication::Get Response [C0002.017]: nhận phản hồi HTTP.
  + HTTP Communication::Read Header [C0002.014]: đọc phần header của phản hồi HTTP.
  + HTTP Communication::Send Request [C0002.003]: gửi yêu cầu HTTP.
  + Socket Communication::Get Socket Status [C0001.012]: kiểm tra trạng thái của socket.
  + Socket Communication::Initialize Winsock Library [C0001.009]: khởi tạo thư viện Winsock.
  + Socket Communication::Receive Data [C0001.006]: nhận dữ liệu thông qua socket.
  + Socket Communication::Send Data [C0001.007]: gửi dữ liệu thông qua
  + socket. Socket Communication::Set Socket Config [C0001.001]: cấu hình socket.
* **CRYPTOGRAPHY :** 
  + Cryptographic Hash: đây là phương pháp để tạo ra một băm (hash) duy nhất cho một chuỗi dữ liệu. Phương thức này được định nghĩa trong mã C0029.
  + Cryptographic Hash::MD5: đây là phương thức sử dụng thuật toán MD5 để tạo ra một băm cho một chuỗi dữ liệu. Phương thức này được định nghĩa trong mã C0029.001.
  + Encrypt Data: đây là phương pháp mã hóa dữ liệu bằng cách sử dụng một khóa mã hóa. Phương thức này được định nghĩa trong mã C0027.
  + Encrypt Data::RC4: đây là phương pháp sử dụng thuật toán RC4 để mã hóa dữ liệu với một khóa mã hóa. Phương thức này được định nghĩa trong mã C0027.009.
  + Generate Pseudo-random Sequence: đây là phương pháp tạo ra một chuỗi số ngẫu nhiên sử dụng một giá trị khởi tạo. Phương thức này được định nghĩa trong mã C0021.
  + Generate Pseudo-random Sequence::RC4 PRGA: đây là phương pháp sử dụng thuật toán RC4 để tạo ra một chuỗi số ngẫu nhiên với một giá trị khởi tạo. Phương thức này được định nghĩa trong mã C0021.004
* **DATA**
  + Checksum::CRC32 là một thuật toán kiểm tra tính toàn vẹn dữ liệu bằng cách tính toán và so sánh giá trị CRC (Cyclic Redundancy Check) của dữ liệu.
  + Encode Data::XOR là một phương pháp mã hóa dữ liệu bằng phép XOR (Exclusive OR).
* **DEFENSE EVASION** 
  + Obfuscated Files or Information::Encoding-Standard Algorithm [E1027.m02] : sử dụng các thuật toán mã hóa chuẩn (ví dụ: mã hóa Base64) để giấu đi nội dung
* **DISCOVERY**
  + DISCOVERY | File and Directory Discovery [E1083]: Phát hiện các tệp tin và thư mục trên hệ thống.
  + System Information Discovery [E1082]: Phát hiện thông tin hệ thống.
* **EXECUTION**
  + "Command and Scripting Interpreter" : phương pháp trong giai đoạn "Execution" của một cuộc tấn công. Nó cho phép kẻ tấn công thực thi các lệnh và mã độc thông qua việc sử dụng các trình thông dịch lệnh hoặc script để thực thi các lệnh tại đầu dòng lệnh hoặc tệp script. Các tệp script thường được sử dụng để tổ chức các lệnh thành các tập lệnh phức tạp hơn, hoặc để chạy các lệnh đó một cách tự động.
* **FILE SYSTEM**: các hành động đối với filesystem
  + Tạo thư mục (Create Directory)
  + Xóa thư mục (Delete Directory)
  + Xóa tệp (Delete File)
  + Lấy thuộc tính tệp (Get File Attributes)
  + Di chuyển tệp (Move File)
  + Đọc tệp (Read File)
  + Thiết lập thuộc tính tệp (Set File Attributes)
  + Viết tệp (Write File)
* **MEMORY** 
  + Allocate Memory [C0007] là yêu cầu bộ nhớ của hệ thống để dành một khu vực nhớ cho quá trình thực thi của chương trình độc hại.
* **OPERATING SYSTEM**
  + Registry::Delete Registry Key [C0036.002]: xóa một khóa đăng ký khỏi registry của hệ thống.
  + Registry::Delete Registry Value [C0036.007]: xóa một giá trị đăng ký khỏi registry của hệ thống
  + Registry::Query Registry Key [C0036.005]: truy vấn thông tin của một khóa đăng ký trong registry của hệ thống.
  + Registry::Query Registry Value [C0036.006]: truy vấn thông tin của một giá trị đăng ký trong registry của hệ thống.
  + Registry::Set Registry Key [C0036.001]: tạo một khóa đăng ký mới trong registry của hệ thống.
* **PROCESS**:
  + Allocate Thread Local Storage [C0040]: yêu cầu một không gian nhớ được phân bổ cho luồng cục bộ của một tiến trình.
  + Check Mutex [C0043]: kiểm tra trạng thái của mutex.
  + Create Mutex [C0042]: tạo một mutex mới.
  + Create Process [C0017]: tạo một tiến trình mới.
  + Create Thread [C0038]: tạo một luồng mới trong một tiến trình đã tồn tại.
  + Enumerate Threads [C0064]: liệt kê các luồng trong một tiến trình đã tồn tại.
  + Set Thread Local Storage Value [C0041]: thiết lập giá trị của không gian nhớ cục bộ cho một luồng trong một tiến trình.
  + Terminate Process [C0018]: kết thúc một tiến trình đã tồn tại.

A screenshot of a computer

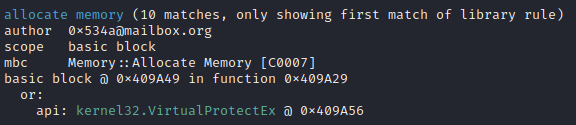
Description automatically generated with medium confidence

* anti-analysis/anti-debugging/debugger-detection: kiểm tra việc trì hoãn thời gian bằng cách sử dụng GetTickCount và phát hiện công cụ phân tích và gỡ lỗi.
* collection/keylog: thu thập thông tin bàn phím bằng cách ghi lại các phím được nhấn. communication: truyền và nhận dữ liệu từ máy chủ từ xa.
* communication/c2/file-transfer: truyền tệp giữa máy khách và máy chủ từ xa.
* communication/dns: giải quyết tên miền DNS.
* communication/http: phân tích
* URL và kiểm tra mã trạng thái HTTP.
* communication/http/client: kết nối với máy chủ HTTP từ xa.
* communication/socket: lấy thông tin và trạng thái kết nối Socket và cấu hình nó.
* data-manipulation/checksum/crc32: tính toán giá trị kiểm tra CRC32 cho dữ liệu.
* data-manipulation/encoding/xor: mã hóa dữ liệu bằng phép XOR. data-
* manipulation/encryption/dpapi: mã hóa dữ liệu bằng DPAPI. data-
* manipulation/encryption/rc4: mã hóa dữ liệu bằng RC4 PRGA. data-
* manipulation/hashing: băm dữ liệu sử dụng MD5 hoặc WinCrypt. data-
* manipulation/prng/lcg: tạo số ngẫu nhiên bằng cách sử dụng LCG trên Delphi.
* data-manipulation/prng/mersenne: tạo số ngẫu nhiên bằng cách sử dụng Mersenne Twister.
* host-interaction/accounts: liệt kê các tài khoản người dùng.
* host-interaction/cli: chấp nhận các tham số dòng lệnh. host-interaction/clipboard: đọc dữ liệu bảng tạm.
* host-interaction/environment-variable: truy vấn biến môi trường.
* host-interaction/file-system: tương tác với hệ thống tệp, bao gồm xóa, tạo, di chuyển và sao chép.
* host-interaction/hardware/storage: lấy thông tin về đĩa.
* host-interaction/mutex: kiểm tra Mutex.
* host-interaction/network/address: lấy địa chỉ IPv4 cục bộ.
* host-interaction/os: tương tác với hệ điều hành, bao gồm tắt hệ thống, lấy tên máy chủ và thông tin hệ thống.
* get hostname: Lấy tên của máy tính trong mạng.
* get system information on Windows: Lấy thông tin hệ thống trên hệ điều hành Windows như tên hệ điều hành, phiên bản hệ điều hành, số lượng bộ nhớ RAM, v.v.
* check OS version: Kiểm tra phiên bản của hệ điều hành đang chạy trên máy tính.
* allocate thread local storage: Phân bổ bộ nhớ cho thread local storage, một kỹ thuật được sử dụng để lưu trữ dữ liệu trong một thread riêng tư.
* get/set thread local storage value: Đọc hoặc ghi giá trị vào thread local storage.
* create process on Windows: Tạo một tiến trình mới trên hệ điều hành Windows.
* allocate RWX memory: Phân bổ bộ nhớ có thể đọc, ghi và thực thi (RWX) để tiêm mã độc vào quá trình hoạt động của một tiến trình.
* enumerate processes: Liệt kê các tiến trình đang chạy trên máy tính.
* modify access privileges: Sửa đổi các quyền truy cập của một tiến trình để thực hiện các hành động như tạo tiến trình mới, tiêm mã độc vào quá trình khác, v.v.
* terminate process: Kết thúc một tiến trình đang chạy trên máy tính. query or
* enumerate registry key/value: Truy vấn hoặc liệt kê các khóa hoặc giá trị trong registry của Windows.
* set/delete registry key/value: Thêm hoặc xóa một khóa hoặc giá trị trong registry của Windows.
* get session user name: Lấy tên người dùng đang đăng nhập vào phiên làm việc của hệ thống.
* create thread: Tạo một thread mới trong tiến trình hiện tại.
* access PEB ldr\_data: Truy cập thông tin lưu trữ về các module được tải trong quá trình của một tiến trình.
* link many functions at runtime: Liên kết nhiều hàm trong thời gian chạy để thực hiện một số hành động nhất định.
* resolve function by hash: Giải quyết các hàm bằng cách sử dụng hàm băm (hash function).
* parse PE header: Phân tích tiêu đề PE (Portable Executable) để tìm thông tin về tệp thực thi.
* "identify system language via API" xác định ngôn ngữ hệ thống thông qua các API được cung cấp bởi hệ điều hành.

1 options nữa của capa là “-vv” làm cho capa không chỉ cho chúng ta thấy những khả năng (capabiliteis) mà mã độc làm được mà nó còn đưa ra lời giải thích tại sao nó nhận định rằng mã độc đó có khả năng như vậy



Ví dụ như 2 đoạn này



Text

Description automatically generated

Ta có thể thấy capa phát hiện đc khả năng của Mã độc dựa vào các api được gọi ở trong file exe

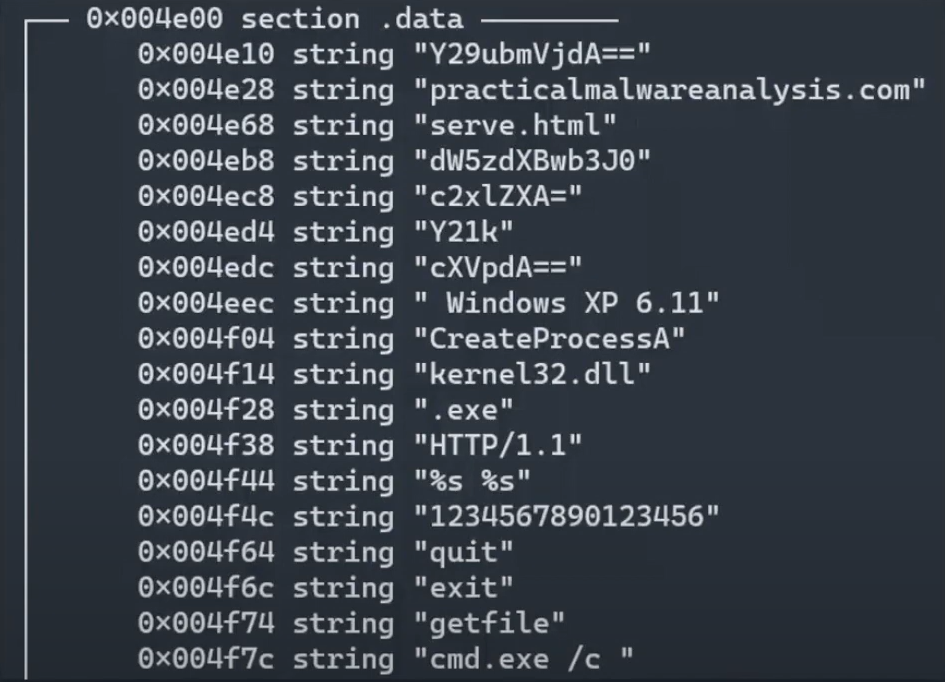
Nó còn có thể chỉ ra chính xác đoạn này nằm ở đâu trong chương trình , điều này khá là quan trọng bởi vì ta có thể đến đoạn đó của chương trình và xem nó có xác định đúng khả năng của mã độc không

Sử dụng IDA pro để xác nhận





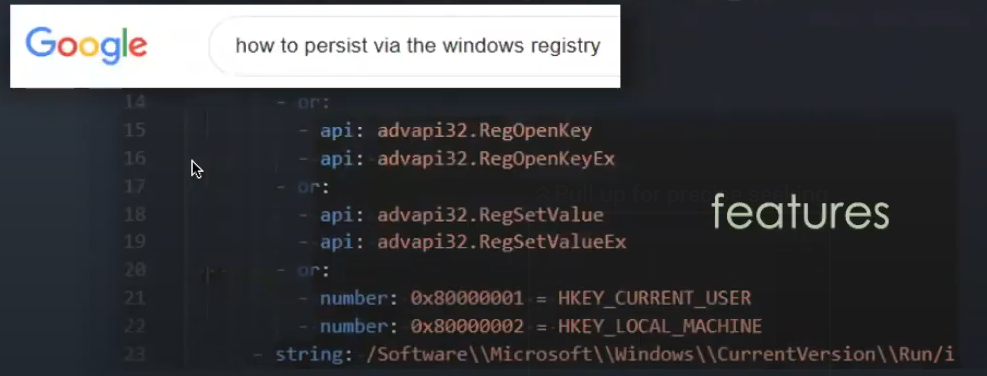
Ngoài ra capa còn có thể trích xuất các file .dll (binary file) thành dạng mà con người có thể đọc hiểu được



Khi đã trích xuất đc dưới dạng human readable thì phát sinh 1 vấn đề nữa là nếu file đc trích xuất quá dài . Vd 10000 dòng thì đối với những người mới phân tích mã độc đây có thể là 1 điều khá là khó khăn vì ta không biết là string nào chúng ta đang cần tìm kiếm và kết hợp các string có liên quan đến nhau để đưa tới kết luận là có mã độc hay không

* Capa có thể giải quyết bằng cách sử dụng các rules của nó và gom nhóm tùy vào từng loại rule
* Không cần đọc từng dòng
* Tập trung khái quát vào data mà capa đưa ra

Ex:



Kết luận :

* Capa là một công cụ phân tích tĩnh mã độc có khả năng tự động phát hiện các khả năng (capabilities) của mã độc dựa trên tệp thực thi của chúng. Với khả năng này, Capa có thể giúp các chuyên gia an ninh mạng và nhà phát triển phần mềm xác định các hành động độc hại có thể thực hiện được bởi mã độc một cách nhanh chóng và tự động.
* Công cụ Capa hoạt động bằng cách phân tích tĩnh tệp thực thi của mã độc và so sánh các tính năng được tìm thấy với một tập hợp các quy tắc và mẫu được định nghĩa sẵn.
* Các khả năng mà Capa có thể phát hiện bao gồm các hành động như Create new process, Write to registry, Create new file, tải thêm mã độc, giả mạo quyền hạn và nhiều hơn nữa.
* Các kết quả phân tích của Capa được hiển thị dưới dạng danh sách **Capabilities**, cùng với thông tin chi tiết về các **Capabilities** đó
  + Chẳng hạn như địa chỉ trong tệp thực thi, API được sử dụng và thông tin thêm về tác giả của khả năng đó.
* Một trong những ưu điểm của Capa là tính tự động hóa cao, giúp đẩy nhanh quá trình phân tích mã độc. Nó cho phép người dùng tập trung vào việc xem xét các khả năng phát hiện được và cải thiện chất lượng của chúng. Ngoài ra, Capa còn hỗ trợ các tập tin định dạng khác nhau, bao gồm tệp PE, ELF và Mach-O.
  + Triage : sàng lọc và phân loại các sự cố hoặc vấn đề dựa trên độ ưu tiên và mức độ nghiêm trọng của chúng , xác định xem một tập tin, hệ thống hoặc mạng có bị nhiễm virus hay không ,..
* Tuy nhiên, việc sử dụng Capa cần có kiến thức chuyên môn về phân tích mã độc và khả năng đánh giá chính xác các kết quả phân tích. Do đó, công cụ này nên được sử dụng bởi các chuyên gia trong lĩnh vực an ninh mạng.
* Tóm lại, Capa là một công cụ rất hữu ích để tự động phát hiện các khả năng của mã độc.

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