ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

Đồ án chuyên ngành

Đề tài:

Trích xuất các thuộc tính tập tin PE phục vụ phát hiện mã độc Windows

Giảng viên hướng dẫn: ThS. Đỗ Thị Thu Hiền , ThS. Nghi Hoàng Khoa

Lớp: **NT114.O21.ANTN - Đồ án chuyên ngành**

Sinh viên thực hiện: Nguyễn Văn Khang Kim 21520314, Nguyễn Vũ Anh Duy 21520211

TP.Hồ Chí Minh, ngày 4 tháng 7 năm 2024

NHẬN XÉT TỪ GIẢNG VIÊN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

MỤC LỤC

MỤC LỤC i
LỜI CẢM ƠN iii
DANH MỤC HÌNH VẼ iv
TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH 1
CHƯƠNG 1: TỔNG QUAN 2
1.1 Giới thiệu 2
1.2 Thách thức và phương pháp 2
1.2.1 Thách thức 2
1.2.2 Phương pháp 2
1.3 Mục tiêu, nội dung cụ thể 2
1.3.1 Mục tiêu 2
1.3.2 Nội dung cụ thể 3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5
2.1 Malware Windows 5
2.2 Machine learning trong việc phân tích và phát hiện malware 5
2.3 Dataset trong Machine learning 6
2.4 Các thuộc tính quan trọng của mã độc 7
2.4.1 Thuộc tính tĩnh 8
2.4.2 Thuộc tính động 9
2.5 Cuckoo sandbox 10
2.5.1 Cuckoo API 10
2.5.2 Hiệu suất của Cuckoo 11

CHƯƠNG 3: CÁCH THỰC HIỆN VÀ KẾT QUẢ THỰC NGHIỆM 12
3.1 Môi trường và công cụ 12
3.2 Tập dữ liệu 12
3.3 Trích xuất thuộc tính tĩnh 13
3.4 Trích xuất thuộc tính động 15
3.4.1 Cài đặt Cuckoo Sandbox 15
3.4.2 Tự động trích xuất với Cuckoo API 16
CHƯƠNG 4: HƯỚNG PHÁT TRIỂN 20
TÀI LIỆU THAM KHẢO 21

LỜI CẢM ƠN

Đầu tiên, em xin chân thành cảm ơn cô Đỗ Thị Thu Hiền và thầy Nghi Hoàng Khoa đã giúp đỡ và hướng dẫn nhóm một cách tận tình trong suốt quá trình thực hiện Đồ án chuyên ngành này.

Em xin chân thành cảm ơn.

TP. Hồ Chí Minh, ngày 04 tháng 07 năm 2024

DANH MỤC HÌNH VẼ

Hình 1: Mô hình hóa công việc 4

Hình 2: Botnet 6

Hình 3: Luồng các mẫu mã độc khi đưa vào Cuckoo 15

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Trong đồ án này, tác giả sẽ tìm cách tìm hiểu cách trích xuất thuộc tính tập tin PE trên Windows.

Trong phạm vi của đồ án này, tác giả sẽ dùng thư viện pefile trong python để trích xuất thuộc tính tĩnh kết hợp với cuckoo sandbox để trích xuất thuộc tính động.

Cuối cùng, tác giả sẽ tự động hóa các bước trích xuất với python.

CHƯƠNG 1: TỔNG QUAN

Trong chương này, tác giả trình bày sơ lược về phương pháp trích xuất thuộc tính tệp tin PE trên Windows và các thách thức mà bài toán đang gặp phải. Tác giả đưa ra mục tiêu, nội dung cụ thể phương pháp thực hiện.

1.1 Giới thiệu

Trong vài năm trở lại đây, mã độc Windows đang ngày càng phổ biến và nguy hiểm hơn, để phát hiện và ngăn chặn mã độc, các phương pháp học máy đang đem lại kết quả tốt. Để phương pháp học máy có thể phát triển hiệu quả và đáng tin cậy, cần phải có một tập dữ liệu đủ lớn và đầy đủ. Bài nghiên cứu này sẽ nghiên cứu về việc trích xuất thuộc tính các tệp PE trong Windows để tạo ra một tập dữ liệu đa dạng về thuộc tính bao gồm cả thuộc tính tĩnh và động.

1.2 Thách thức và phương pháp

1.2.1 Thách thức

Đa số các tập dataset sẵn có chỉ có thuộc tính tĩnh, hoặc là thuộc tính động. Thách thức ở đây là làm sao kết hợp các phương pháp để có thể thu được cả thuộc tính tĩnh và động trên cùng một tệp PE.

1.2.2 Phương pháp

+ Ngôn ngữ: Python là một ngôn ngữ mạnh mẽ giúp tự động hóa các tác vụ và giao tiếp bằng API.
+ Thuộc tính tĩnh: Sử dụng thư viện pefile trong python để thực hiện trích xuất tự động.
+ Thuộc tính động: Sandbox là một giải pháp tuyệt vời đối với việc phân tích mã độc, các thuộc tính động có thể được trích xuất thông qua kết quả báo cáo của sandbox, ở đây tác giả sẽ dùng Cuckoo sandbox.

1.3 Mục tiêu, nội dung cụ thể

1.3.1 Mục tiêu

Tìm hiểu và nghiên cứu phương pháp trích xuất thuộc tính tập tin PE trên Windows.

1.3.2 Nội dung cụ thể

Nội dung 1: Tìm hiểu các thuộc tính thường được sử dụng cho việc phát hiện mã độc Windows.

- Phương pháp thực hiện: Tham khảo các bài báo, tài liệu liên quan. Phân tích đặc điểm của các phương pháp khác nhau để quyết định các thuộc tính được quan tâm.
- Xác định các thuộc tính cũng như công cụ giúp trích xuất các thuộc tính.

Dự kiến kết quả: Nắm vững kiến thức về các thuộc tính cơ bản, chọn được công cụ để sử dụng.

Nội dung 2: Tìm hiểu về sandbox và tự động hóa với python.

Phương pháp thực hiện:

- Tìm hiểu về sandbox và chọn sandbox tối ưu nhất để thực hiện phân tích.
- Tìm hiểu cách sử dụng các thư viện mã nguồn mở của python hỗ trợ trích xuất thuộc tính như pefile ...

Dự kiến kết quả: Tài liệu về cách sử dụng và áp dụng các công nghệ vào phương pháp đã chọn.

Nội dung 3: Xây dựng chương trình trích xuất thuộc tính tĩnh.

Phương pháp thực hiện:

- Extract: Viết chương trình bằng python trích xuất các thuộc tính cụ thể.
- Auto: Tự động hóa trích xuất nhiều thuộc tính.
- Threading: Tăng khả năng trích xuất bằng cách tạo nhiều luồng song song.
- Save: Lưu kết quả vào tệp csv.

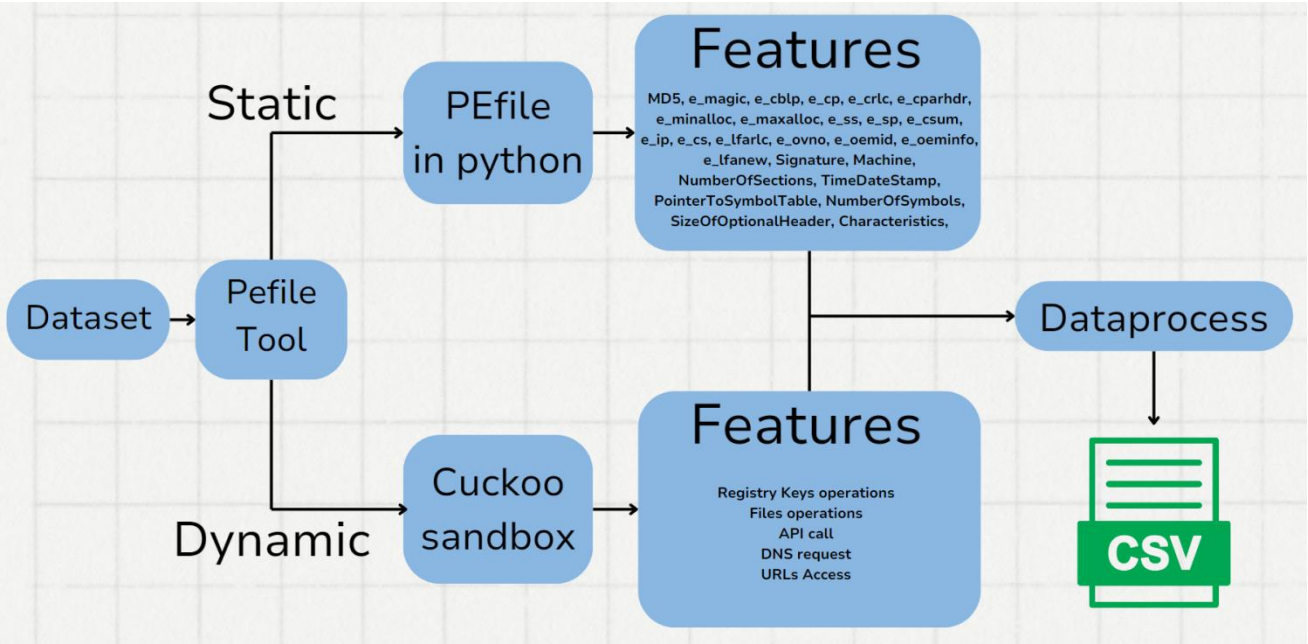
Dự kiến kết quả: Chương trình có khả năng tự động trích xuất nhiều tệp PE file cùng một lúc và kết quả được lưu vào tệp csv.

Nội dung 4: Trích xuất thuộc tính động.

Phương pháp thực hiện:

- Setup: Chuẩn bị môi trường Cuckoo sandbox để thực hiện phân tích, có thể tăng tốc độ xử lý của hộp cát bằng cách thêm nhiều máy ảo cuckoo.
- Communicate: Giao tiếp với cuckoo để thực hiện phân tích động các tệp PE và lấy kết quả báo cáo.
- Extract and save: Trích xuất các thuộc tính động từ báo cáo và lưu kết quả vào tệp csv.

Dự kiến kết quả: Môi trường dùng để phân tích động mã độc, báo cáo phân tích, các thuộc tính trích xuất được từ báo cáo.

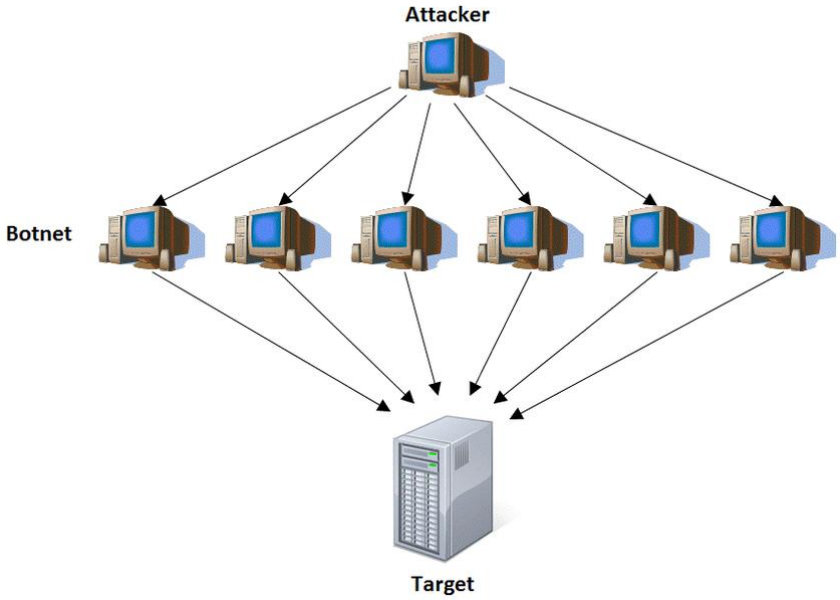


Hình 1. Mô hình hóa các công việc

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Malware Windows

Windows là hệ điều hành gánh chịu sự tấn công của mã độc nặng nề nhất. Mã độc trên Windows rất đa dạng về hình thức và phương pháp lây nhiễm, gây ra nhiều mối đe dọa cho người dùng và hệ thống. Các dạng mã độc phổ biến trên Windows bao gồm virus, Trojan Horse, worm, spyware, adware, ransomware, rootkit, và botnet. Mỗi loại mã độc này có cách thức hoạt động và mục đích khác nhau, nhưng tất cả đều nhằm gây hại hoặc khai thác thông tin từ hệ thống nạn nhân. Virus là một loại mã độc tự gắn vào các chương trình hoặc tập tin khác để lây lan khi các tập tin này được thực thi. Trojan Horse (Trojan) giả dạng thành một chương trình hợp pháp để đánh lừa người dùng cài đặt và thực thi, sau đó thực hiện các hành vi độc hại như đánh cắp thông tin hoặc mở cửa hậu cho hacker. Worm lây lan qua mạng bằng cách tự sao chép mà không cần sự tương tác của người dùng, có thể gây tắc nghẽn mạng và làm suy giảm hiệu suất hệ thống. Spyware thu thập thông tin cá nhân của người dùng mà không được phép, chẳng hạn như theo dõi hoạt động trực tuyến, thu thập dữ liệu nhạy cảm và gửi về máy chủ điều khiển. Adware hiển thị quảng cáo không mong muốn trên máy tính người dùng và có thể đi kèm với spyware để theo dõi hành vi người dùng. Ransomware mã hóa dữ liệu của người dùng và yêu cầu tiền chuộc để khôi phục lại quyền truy cập, gây thiệt hại lớn cho cả cá nhân và tổ chức. Rootkit giúp kẻ tấn công ẩn mình trong hệ thống và duy trì quyền truy cập mà không bị phát hiện. Botnet là một mạng lưới các máy tính bị nhiễm mã độc, được điều khiển từ xa bởi hacker để thực hiện các cuộc tấn công từ chối dịch vụ (DDoS), spam, hoặc các hoạt động bất hợp pháp khác. Các phương pháp lây nhiễm của mã độc Windows rất đa dạng. Một số phương pháp thông thường bao gồm việc phát tán qua email với các tệp tin đính kèm độc hại hoặc các liên kết trong email lừa đảo. Người dùng cũng có thể nhiễm mã độc khi tải và cài đặt phần mềm từ các nguồn không đáng tin cậy trên Internet. Thiết bị USB cũng là một phương tiện lây nhiễm phổ biến khi các thiết bị lưu trữ di động đã bị nhiễm được cắm vào máy tính. Ngoài ra, mã độc có thể tấn công vào các lỗ hổng bảo mật trong hệ điều hành hoặc phần mềm không được vá lỗi kịp thời, khai thác các điểm yếu này để xâm nhập và gây hại cho hệ thống.



Hình 2. Botnet

2.2 Machine learning trong việc phân tích và phát hiện malware.

Machine Learning (ML) đang ngày càng được ứng dụng rộng rãi trong lĩnh vực an ninh mạng, đặc biệt là trong việc phân tích và phát hiện malware. ML cung cấp các phương pháp tiên tiến giúp phân tích các mẫu mã độc phức tạp và không ngừng thay đổi, từ đó cải thiện hiệu quả và độ chính xác trong việc phát hiện mã độc.

Trong phân loại malware, ML có thể được áp dụng thông qua hai phương pháp chính: phân tích tĩnh và phân tích động. Phân tích tĩnh sử dụng các đặc trưng không thay đổi của tập tin, chẳng hạn như mã nhị phân, các chuỗi ký tự, và các hàm gọi hệ thống. Các mô hình học máy có thể học từ những đặc trưng này để phân loại tập tin thành mã độc hoặc mã lành. Mặt khác, phân tích động tập trung vào việc quan sát hành vi của mã độc khi nó được thực thi trong môi trường sandbox. ML có thể học từ các hành vi này để phát hiện và phân loại các mẫu mã độc tương tự trong tương lai.

Phát hiện malware cũng là một lĩnh vực mà ML thể hiện sự hiệu quả vượt trội, đặc biệt là trong việc phát hiện các hành vi bất thường (anomaly detection). ML có khả năng học từ các mẫu hành vi thông thường của hệ thống để nhận diện khi có bất kỳ hoạt động nào lệch khỏi

chuẩn mực. Các mô hình học máy có thể phân tích lưu lượng mạng, hoạt động hệ thống, và các sự kiện khác để phát hiện các mẫu mã độc mới và chưa từng được biết đến trước đó. Điều này đặc biệt quan trọng trong việc chống lại các cuộc tấn công từ mã độc zero-day, khi mà các biện pháp bảo mật truyền thống có thể không đủ khả năng phát hiện.

Một ứng dụng khác của ML trong phân tích malware là phân tích các mối quan hệ và sự tương đồng giữa các mẫu mã độc. Các thuật toán học máy có thể xác định các đặc trưng chung giữa các mẫu mã độc khác nhau, giúp các nhà nghiên cứu nhận diện các chiến dịch tấn công phức tạp và các họ mã độc có liên quan. Điều này không chỉ giúp cải thiện khả năng phát hiện mà còn cung cấp thông tin quan trọng để phát triển các biện pháp phòng ngừa và phản ứng hiệu quả hơn.

Sử dụng Machine Learning trong phân tích malware không chỉ cải thiện hiệu quả phát hiện và phân loại mã độc mà còn giúp các hệ thống bảo mật thích ứng nhanh chóng với các mối đe dọa mới. ML không ngừng học hỏi và cải thiện từ các dữ liệu mới, giúp các công cụ bảo mật trở nên linh hoạt và mạnh mẽ hơn trong cuộc chiến chống lại mã độc.

2.3 Dataset trong Machine learning

Dataset là yếu tố cực kỳ quan trọng trong việc phát triển các mô hình Machine Learning (ML) cho phân tích và phát hiện mã độc. Một dataset chất lượng cao sẽ giúp mô hình ML học được các đặc trưng chính xác của mã độc, từ đó cải thiện khả năng phát hiện và phân loại mã độc. Để đạt hiệu quả tốt nhất, dataset cần bao gồm nhiều loại mã độc khác nhau như virus, trojan, worm, ransomware, spyware, và adware, nhằm đảm bảo mô hình ML có thể nhận diện và phân biệt được nhiều loại mã độc khác nhau và không bị chênh lệch. Ngoài sự đa dạng về loại mã độc, dataset cũng cần có số lượng mẫu đủ lớn để mô hình ML có thể học một cách hiệu quả và tránh bị overfitting. Số lượng mẫu lớn giúp mô hình nắm bắt được các đặc trưng quan trọng và tăng khả năng tổng quát hóa. Thêm vào đó, dataset cần bao gồm cả mã lành (benign) để mô hình có thể học cách phân biệt giữa mã độc và mã lành, từ đó giảm thiểu tỷ lệ báo động giả (false positives). Một yếu tố quan trọng khác là chất lượng và tính đại diện của dữ liệu. Dataset cần phản ánh đúng thực tế các mẫu mã độc gặp phải trong môi trường thực tế. Điều này bao gồm việc cập nhật dataset thường xuyên để bao gồm các mẫu mã độc mới và các biến

thể của chúng. Các dữ liệu trong dataset cần được gắn nhãn chính xác để đảm bảo rằng mô hình học từ dữ liệu đáng tin cậy và đúng đắn. Cuối cùng, dataset cần được chuẩn bị và xử lý một cách cẩn thận trước khi được đưa vào huấn luyện mô hình ML. Điều này bao gồm việc loại bỏ dữ liệu nhiễu, chuẩn hóa dữ liệu, và chia tách dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu quả của mô hình một cách khách quan. Một dataset được chuẩn bị kỹ lưỡng sẽ là nền tảng vững chắc cho việc phát triển các mô hình ML mạnh mẽ và hiệu quả trong việc phát hiện và phân tích mã độc.

2.4 Các thuộc tính quan trọng của mã độc.

2.4.1 Thuộc tính tĩnh.

Khi phân tích mã độc bằng Machine Learning, các thuộc tính tĩnh đóng vai trò quan trọng trong việc xác định đặc điểm và hành vi của mã độc mà không cần thực thi mã. Các thuộc tính tĩnh này được tham khảo từ nhiều bài báo khác nhau về chủ đề ML trong việc phát hiện mã độc Windows bao gồm 82 thuộc tính quan trọng như:

Name, MD5, e_magic, e_cblp, e_cp, e_crlc, e_cparhdr, e_minalloc, e_maxalloc, e_ss, e_sp, e_csum, e_ip, e_cs, e_lfarlc, e_ovno, e_oemid, e_oeminfo, e_lfanew, Signature, Machine, NumberOfSections, TimeDateStamp, PointerToSymbolTable, NumberOfSymbols, SizeOfOptionalHeader, Characteristics, Magic, MajorLinkerVersion, MinorLinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion, MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, Reserved1, SizeOfImage, SizeOfHeaders, CheckSum, Subsystem, DllCharacteristics, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, LoaderFlags, NumberOfRvaAndSizes, SectionsNb, SectionsMeanEntropy, SectionsMinEntropy, SectionsMaxEntropy, CharacteristicsMean, CharacteristicsMin, CharacteristicsMax, SectionsMeanRawsize, SectionsMinRawsize, SectionMaxRawsize, SectionsMeanVirtualsize, SectionsMinVirtualsize, SectionMaxVirtualsize, ImportsNbDLL, ImportsNb, ImportsNbOrdinal, ExportNb, ResourcesNb, ResourcesMeanEntropy, ResourcesMinEntropy,

ResourcesMaxEntropy, ResourcesMeanSize, ResourcesMinSize, ResourcesMaxSize, LoadConfigurationSize, và VersionInformationSize.

Những thuộc tính này cung cấp thông tin chi tiết về cấu trúc, kích thước, phiên bản và các đặc điểm khác của tập tin, giúp các mô hình Machine Learning phân tích và phát hiện mã độc một cách hiệu quả mà không cần thực thi trực tiếp mã độc trên hệ thống.

2.4.2 Thuộc tính động.

Thuộc tính động của mã độc là các đặc điểm được thu thập khi mã độc được thực thi trong một môi trường giám sát, chẳng hạn như sandbox. Những thuộc tính này giúp xác định hành vi thực tế của mã độc và cung cấp thông tin chi tiết về cách mã độc tương tác với hệ thống và mạng lưới. Một số thuộc tính động quan trọng bao gồm:

Registry Keys operations: Thuộc tính này chứa thông tin về các thay đổi hoặc truy cập vào sổ đăng ký Windows. Bao gồm các hành vi như ghi sổ đăng ký, xóa sổ đăng ký, mở và đọc các khóa đăng ký. Việc mã độc tương tác với registry keys có thể tiết lộ nhiều về các thiết lập hệ thống mà mã độc cố gắng thay đổi hoặc thao túng để duy trì sự hiện diện của nó hoặc để thực hiện các hành vi độc hại.

Files operations: Thuộc tính này ghi lại các hoạt động liên quan đến tệp tin như tạo, sửa đổi, xóa và số lượng tệp bị lỗi. Việc theo dõi các thay đổi đối với hệ thống tệp có thể cung cấp manh mối về hành vi của mã độc, chẳng hạn như việc mã độc cố gắng sao chép chính nó vào nhiều vị trí khác nhau, hoặc sửa đổi các tệp hệ thống quan trọng để thao túng hệ thống.

API call: API (Application Programming Interface) là tập hợp các lệnh gọi chương trình con hoặc hàm được sử dụng để giao tiếp giữa các thành phần phần mềm hoặc giữa phần mềm và phần cứng. Thuộc tính này ghi lại các cuộc gọi API mà mã độc thực hiện, cung cấp thông tin về các chức năng mà mã độc sử dụng để thực hiện các hành vi của nó. Ví dụ, các cuộc gọi API để truy cập mạng, thao tác tệp, hoặc thay đổi các thiết lập hệ thống.

DNS request: Thuộc tính này ghi lại số lượng truy vấn DNS mà mã độc thực hiện. Truy vấn DNS được sử dụng để chuyển đổi các tên miền thành các địa chỉ IP và có thể tiết lộ các máy

chủ mà mã độc cố gắng liên lạc. Việc theo dõi các truy vấn DNS có thể giúp xác định các kết nối tới máy chủ điều khiển và chỉ huy (C2) của mã độc.

URLs access: Thuộc tính này ghi lại số lượng các URL mà mã độc truy cập. Việc theo dõi các URL được truy cập có thể cung cấp thông tin về các trang web hoặc tài nguyên trực tuyến mà mã độc cố gắng tải về hoặc gửi dữ liệu đến. Điều này đặc biệt quan trọng trong việc phát hiện các hành vi như tải về mã độc bổ sung, truyền tải dữ liệu bị đánh cắp, hoặc liên lạc với các máy chủ điều khiển.

Các thuộc tính động này là một phần không thể thiếu trong việc phân tích và phát hiện mã độc bằng Machine Learning. Chúng cung cấp cái nhìn sâu sắc về hành vi thực tế của mã độc, giúp các mô hình Machine Learning học được các đặc trưng quan trọng và nâng cao khả năng phát hiện các mẫu mã độc mới và chưa được biết đến.

2.5 Cuckoo Sandbox.

Cuckoo Sandbox là một hệ thống phân tích mã độc mã nguồn mở mạnh mẽ, được thiết kế để tự động phân tích các tập tin đáng ngờ trong các môi trường ảo hóa. Cuckoo Sandbox cho phép người dùng thực thi và theo dõi hành vi của mã độc trong một môi trường cô lập, thu thập thông tin chi tiết về hành vi của mã độc và các tác động của nó đến hệ thống. Hệ thống này tự động phân tích các tập tin đáng ngờ, bao gồm tệp thực thi, tài liệu, script và nhiều loại tệp khác. Nó sử dụng các công nghệ ảo hóa như VirtualBox, VMware, KVM, và Xen để tạo ra môi trường phân tích cô lập, đảm bảo rằng mã độc không thể ảnh hưởng đến hệ thống thực tế. Cuckoo Sandbox theo dõi và ghi lại mọi hành vi của mã độc, bao gồm các thay đổi đến hệ thống tệp, registry, các cuộc gọi API, truy vấn DNS, và các hoạt động mạng khác.

2.5.1 Cuckoo API.

Cuckoo API là một giao diện lập trình ứng dụng mạnh mẽ và linh hoạt cho phép người dùng tương tác và điều khiển Cuckoo Sandbox từ các ứng dụng bên ngoài. Thông qua Cuckoo API, người dùng có thể thực hiện các tác vụ như gửi tệp tin để phân tích, truy xuất kết quả phân tích, và quản lý các mẫu mã độc một cách dễ dàng và hiệu quả. API cung cấp các phương thức để tải lên tệp tin đáng ngờ, khởi động quá trình phân tích, và lấy các báo cáo chi tiết về hành vi của mã độc, bao gồm các hoạt động mạng và các thay đổi hệ thống. Ngoài ra, Cuckoo

API còn cho phép người dùng cấu hình và tùy chỉnh Cuckoo Sandbox, tích hợp với các hệ thống giám sát bảo mật và các nền tảng phân tích mã độc tự động. Bằng cách sử dụng Cuckoo API, các tổ chức và cá nhân có thể nâng cao khả năng phát hiện và phân tích mã độc, đảm bảo an toàn và bảo mật cho hệ thống của họ.

2.5.2 Hiệu suất của Cukoo.

Không giống như phân tích tĩnh, hiệu suất của phân tích động phụ thuộc vào khả năng phân tích mẫu của Cuckoo Sandbox. Tốc độ phân tích của Cuckoo Sandbox chủ yếu phụ thuộc vào hai yếu tố chính: số lượng máy ảo được sử dụng và khả năng tạo báo cáo của máy chủ Cuckoo. Để nâng cao hiệu suất, một trong những cách đơn giản và hiệu quả nhất chính là tạo nhiều máy ảo hơn. Việc tạo nhiều máy ảo cho phép hệ thống thực hiện phân tích song song, giảm thời gian chờ đợi và tăng khả năng xử lý nhiều mẫu mã độc cùng lúc. Điều này không chỉ tối ưu hóa việc sử dụng tài nguyên hệ thống mà còn giúp cải thiện hiệu quả và tốc độ phân tích tổng thể của Cuckoo Sandbox.

CHƯƠNG 3: CÁCH THỰC HIỆN VÀ KẾT QUẢ THỰC NGHIỆM

3.1 Môi trường và công cụ.

Môi trường: Máy ảo Linux, RAM 8GB , 2 CPU, Memory 100GB.

Công cụ: Cuckoo Sanbox, pefile.

Ngôn ngữ: Python2.7.

3.2 Tập dữ liệu

- Dataset (10690)
  - Benign (1000)
    - Benign Test (300)
    - Benign Train (700)
  - Virus (9690)
    - Virus Test (2691)
      - Locker(99)
      - Mediyes(435)
      - Winwebsec(1320)
      - Zbot(630)
      - Zeroaccess(207)
    - Virus Train (6999)
      - Locker(231)
      - Mediyes(1015)
      - Winwebsec(3080)
      - Zbot(1470)

- Zeroaccess(483)

3.3 Trích xuất thuộc tính tĩnh.

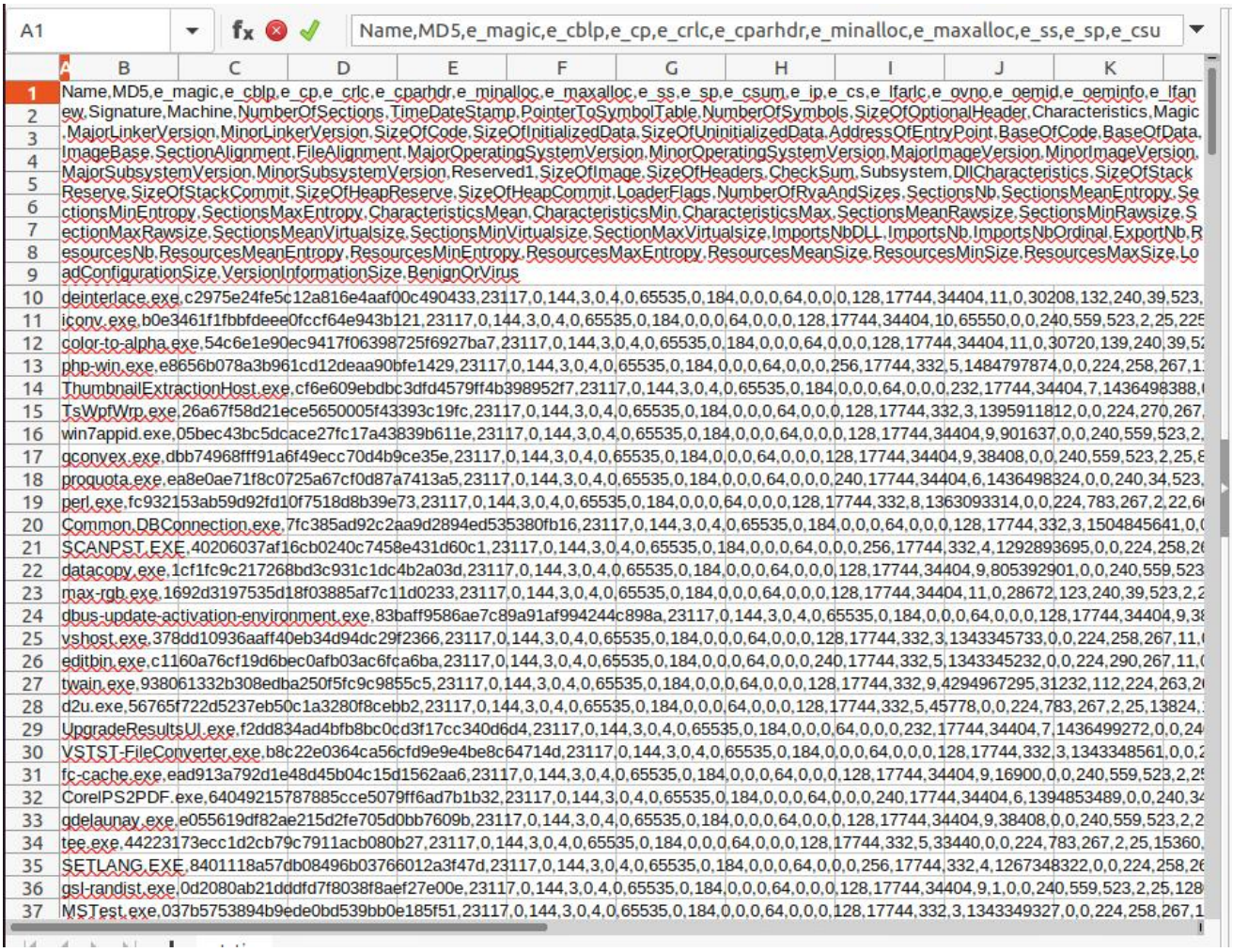
Để thực hiện phân tích 82 thuộc tính tĩnh, thư viện chủ yếu được dùng là pefile trong Python. Các thuộc tính trong *Dos Header, NT Header, File Header, Optional Header* đều có thể trích xuất trực tiếp bằng thư viện pefile. Các entropy sẽ được tính toán theo cách của Shannon (Shannon entropy) và sẽ dùng để tính toán các thuộc tính liên quan đến entropy. Các giá trị *mean, min* và *max* sẽ được tính cho *entropy, characteristics, raw_size, vitural_size* của *Sections* và *entropy, size* của *Resources.* Các thuộc tính *Load_Configuration_Size* và *Version_Information_Size* sẽ được trích xuất trực tiếp từ tệp PE, *Md5* sẽ được tính bằng hàm có sẵn của Python.

Để tăng tốc độ xử lý, Threading là một kĩ thuật tuyệt vời. Với 10 luồng cùng một lúc, thời gian trích xuất 300 tệp sẽ được rút ngắn thành 5-7 phút.

Thực hiện phân tích tĩnh với 300 mẫu của Benign Test.

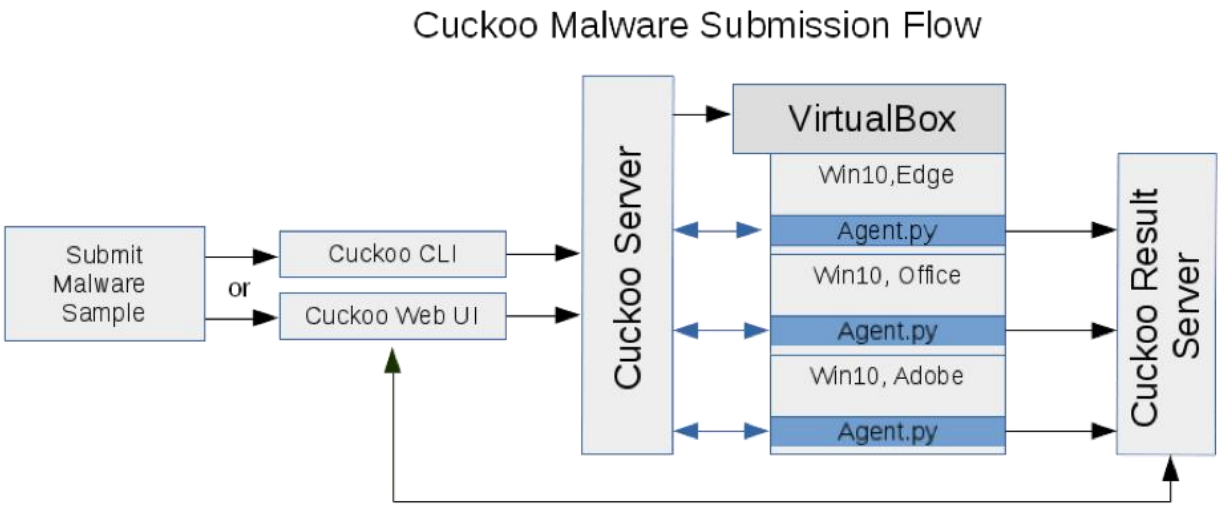
```
khangkim@khangkimvm:~/Desktop$ python static_v1.py -s Dataset/Benign/Benign_test/ -d ./ -bov 0
khangkim@khangkimvm:~/Desktop$
```

Kết quả thu được bao gồm tất cả thuộc tính tập tin PE nằm trong tệp static.csv.


A1
Name,MD5,e_magic,e_cblp,e_cp,e_crlc,e_cparhdr,e_minalloc,e_maxalloc,e_ss,e_sp,e_csu
1 Name,MD5,e_magic,e_cblp,e_cp,e_crlc,e_cparhdr,e_minalloc,e_maxalloc,e_ss,e_sp,e_csum,e_ip,e_cs,e_lfarlc,e_ovno,e_oemid,e_oeminfo,e_lfanew,Signature,Machine,NumberOfSections,TimeDateStamp,PointerToSymbolTable,NumberOfSymbols,SizeOfOptionalHeader,Characteristics,Magic,MajorLinkerVersion,MinorLinkerVersion,SizeOfCode,SizeOfInitializedData,SizeOfUninitializedData,AddressOfEntryPoint,BaseOfCode,BaseOfData,ImageBase,SectionAlignment,FileAlignment,MajorOperatingSystemVersion,MinorOperatingSystemVersion,MajorImageVersion,MinorImageVersion,MajorSubsystemVersion,MinorSubsystemVersion,Reserved1,SizeOfImage,SizeOfHeaders,CheckSum,Subsystem,DllCharacteristics,SizeOfStackReserve,SizeOfStackCommit,SizeOfHeapReserve,SizeOfHeapCommit,LoaderFlags,NumberOfRvaAndSizes,SectionsNb,SectionsMeanEntropy,SectionsMinEntropy,SectionsMaxEntropy,CharacteristicsMean,CharacteristicsMin,CharacteristicsMax,SectionsMeanRawsize,SectionsMinRawsize,SectionMaxRawsize,SectionsMeanVirtualsize,SectionsMinVirtualsize,SectionMaxVirtualsize,ImportsNbDLL,ImportsNb,ImportsNbOrdinal,ExportNb,ResourcesNb,ResourcesMeanEntropy,ResourcesMinEntropy,ResourcesMaxEntropy,ResourcesMeanSize,ResourcesMinSize,ResourcesMaxSize,LoadConfigurationSize,VersionInformationSize,BenignOrVirus
10 deinterlace.exe,c2975e24fe5c12a816e4aaf00c490433,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,11,0,30208,132,240,39,523,
11 iconv.exe,b0e3461f1fbbfdeee0fccf64e943b121,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,10,65550,0,0,240,559,523,2,25,225
12 color-to-alpha.exe,54c6e1e90ec9417f06398725f6927ba7,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,11,0,30720,139,240,39,52
13 php-win.exe,e8656b078a3b961cd12deaa90bfe1429,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,256,17744,332,5,1484797874,0,0,224,258,267,1
14 ThumbnailExtractionHost.exe,cf6e609ebdbc3dfd4579ff4b398952f7,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,232,17744,34404,7,1436498388,
15 TsWpfWrp.exe,26a67f58d21ece5650005f43393c19fc,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,3,1395911812,0,0,224,270,267,
16 win7appid.exe,05bec43bc5dcace27fc17a43839b611e,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,901637,0,0,240,559,523,2,
17 gconvex.exe,dbb74968fff91a6f49ecc70d4b9ce35e,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,38408,0,0,240,559,523,2,25,
18 proquota.exe,ea8e0ae71f8c0725a67cf0d87a7413a5,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,240,17744,34404,6,1436498324,0,0,240,34,523,
19 perl.exe,fc932153ab59d92fd10f7518d8b39e73,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,8,1363093314,0,0,224,783,267,2,22,6
20 Common.DBConnection.exe,7fc385ad92c2aa9d2894ed535380fb16,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,3,1504845641,0,
21 SCANPST.EXE,40206037af16cb0240c7458e431d60c1,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,256,17744,332,4,1292893695,0,0,224,258,2
22 datacopy.exe,1cf1fc9c217268bd3c931c1dc4b2a03d,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,805392901,0,0,240,559,523
23 max-rgb.exe,1692d3197535d18f03885af7c11d0233,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,11,0,28672,123,240,39,523,2,2
24 dbus-update-activation-environment.exe,83baff9586ae7c89a91af994244c898a,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,3
25 vshost.exe,378dd10936aaff40eb34d94dc29f2366,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,3,1343345733,0,0,224,258,267,11,
26 editbin.exe,c1160a76cf19d6bec0afb03ac6fca6ba,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,240,17744,332,5,1343345232,0,0,224,290,267,11,
27 twain.exe,938061332b308edba250f5fc9c9855c5,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,9,4294967295,31232,112,224,263,2
28 d2u.exe,56765f722d5237eb50c1a3280f8cebb2,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,5,45778,0,0,224,783,267,2,25,13824,
29 UpgradeResultsUI.exe,f2dd834ad4bfb8bc0cd3f17cc340d6d4,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,232,17744,34404,7,1436499272,0,0,24
30 VSTST-FileConverter.exe,b8c22e0364ca56cfd9e9e4be8c64714d,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,3,1343348561,0,0,2
31 fc-cache.exe,ead913a792d1e48d45b04c15d1562aa6,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,16900,0,0,240,559,523,2,25
32 CorelPS2PDF.exe,64049215787885cce5079ff6ad7b1b32,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,240,17744,34404,6,1394853489,0,0,240,34
33 qdelaunay.exe,e055619df82ae215d2fe705d0bb7609b,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,38408,0,0,240,559,523,2,2
34 tee.exe,44223173ecc1d2cb79c7911acb080b27,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,5,33440,0,0,224,783,267,2,25,15360,
35 SETLANG.EXE,8401118a57db08496b03766012a3f47d,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,256,17744,332,4,1267348322,0,0,224,258,26
36 gsl-randist.exe,0d2080ab21dddfd7f8038f8aef27e00e,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,34404,9,1,0,0,240,559,523,2,25,128
37 MSTest.exe,037b5753894b9ede0bd539bb0e185f51,23117,0,144,3,0,4,0,65535,0,184,0,0,0,64,0,0,0,128,17744,332,3,1343349327,0,0,224,258,267,1

3.4 Trích xuất thuộc tính động

3.4.1 Cài đặt Cuckoo Sandbox.

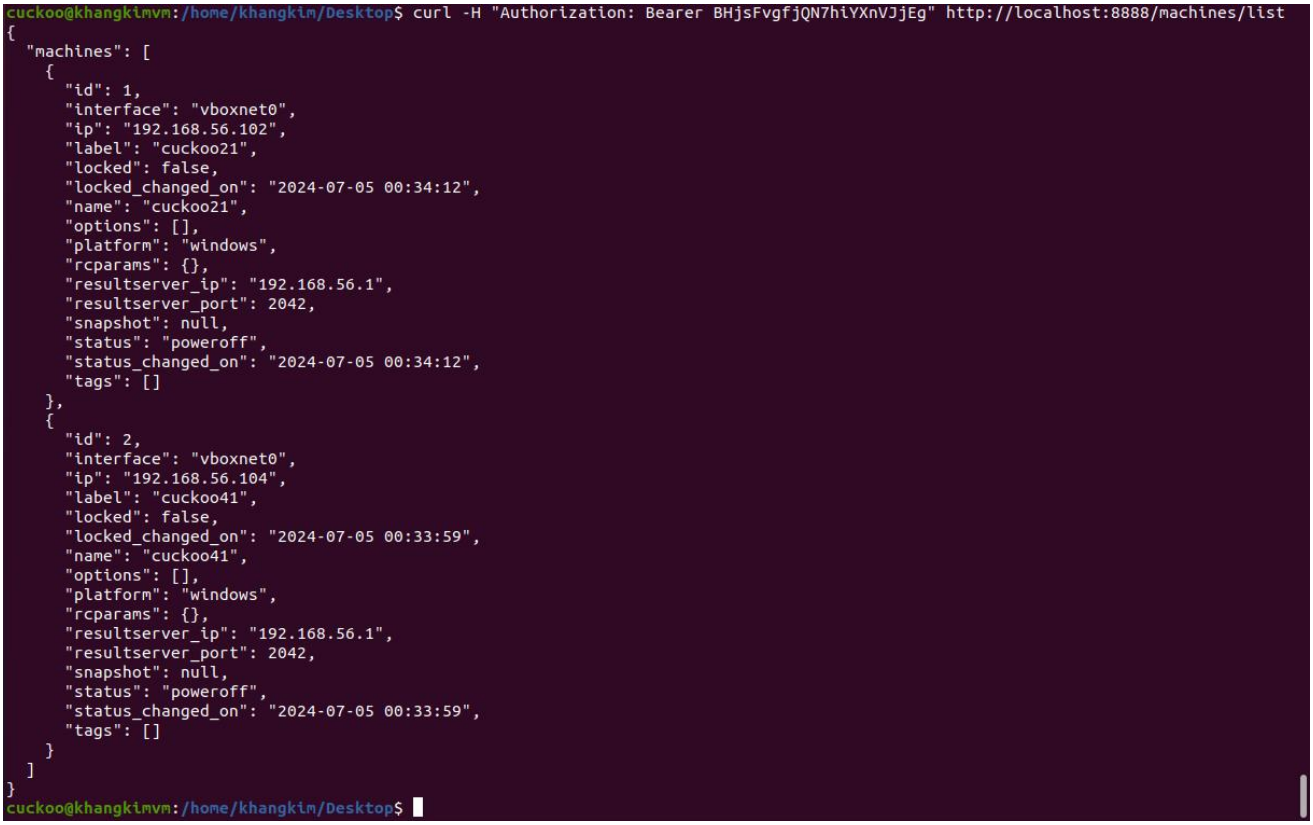


Hình 2. Luồng các mẫu mã độc khi đưa vào Cuckoo.

Ta có thể gửi và phân tích các mẫu cho Cuckoo bằng giao diện web(Web UI) hoặc sửa dụng Cuckoo command line interface (hay còn được gọi là Cuckoo API). Để sửa dụng Cuckoo API, ta sẽ khởi động API server tại local host 127.0.0.1 và port 8888.

```
(cuckoo) cuckoo@khangkimvm:~/.cuckoo$ cuckoo api --host 127.0.0.1 --port 8888
/home/cuckoo/.virtualenvs/cuckoo/lib/python2.7/site-packages/sflock/decode/office.py:12: CryptographyDeprecationWarning: Python 2 is no
 longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
2024-07-04 21:17:30,041 [werkzeug] INFO:  * Running on http://127.0.0.1:8888/ (Press CTRL+C to quit)
```

Số lượng máy ảo chính là số lượng các mẫu sẽ được xử lý cùng lúc. Vì lo ngại về phần cứng không đáp ứng được quá nhiều máy ảo cùng lúc nên ta chỉ tạo 2 máy ảo RAM 2GB.



3.4.2 Tự động trích xuất với Cuckoo API.

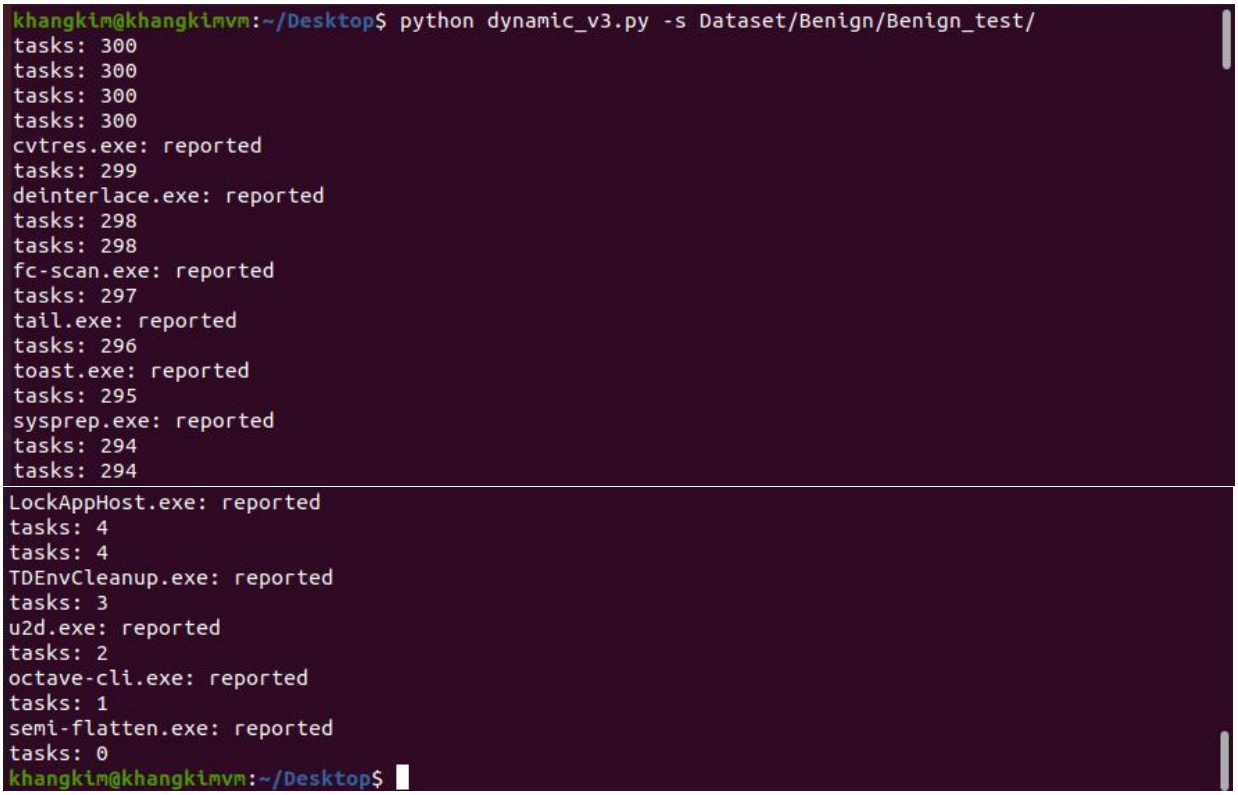
Để tự động hóa việc trích xuất với Cuckoo API, ý tưởng ở đây là viết một mã Python có chức năng gửi tất cả các các mẫu lên hàng đợi của Cuckoo cùng lúc, hỏi Cuckoo trạng thái của các mẫu định kỳ và thực hiện lấy báo cáo, trích xuất những mẫu đã hoàn thành báo cáo, cuối cùng xóa các mẫu này ra khỏi cơ sở dữ liệu để đỡ gánh nặng cho bộ nhớ.

Các API được sử dụng:

- /tasks/create/file: Thêm một mẫu vào danh sách các nhiệm vụ đang chờ xử lý. Trả về ID của tác vụ mới được tạo. Sử dụng vòng lặp và gửi tất cả các mẫu cùng một lúc, đây là bước đầu tiên của quá trình trích xuất.
- /tasks/list: Trả về danh sách các nhiệm vụ. Dùng API này để lấy trạng thái của các mẫu, nó cho ta biết mẫu nào chưa phân tích, đã phân tích xong, và đã tạo xong báo cáo. Dùng vòng lặp và sleep để lặp lại sau mỗi 10 giây.
- /tasks/sample: Trả về danh sách nhiệm vụ cho mẫu. API này chỉ sử dụng khi muốn lấy trạng thái của một mẫu bất kì dựa trên ID liên kết với nó.

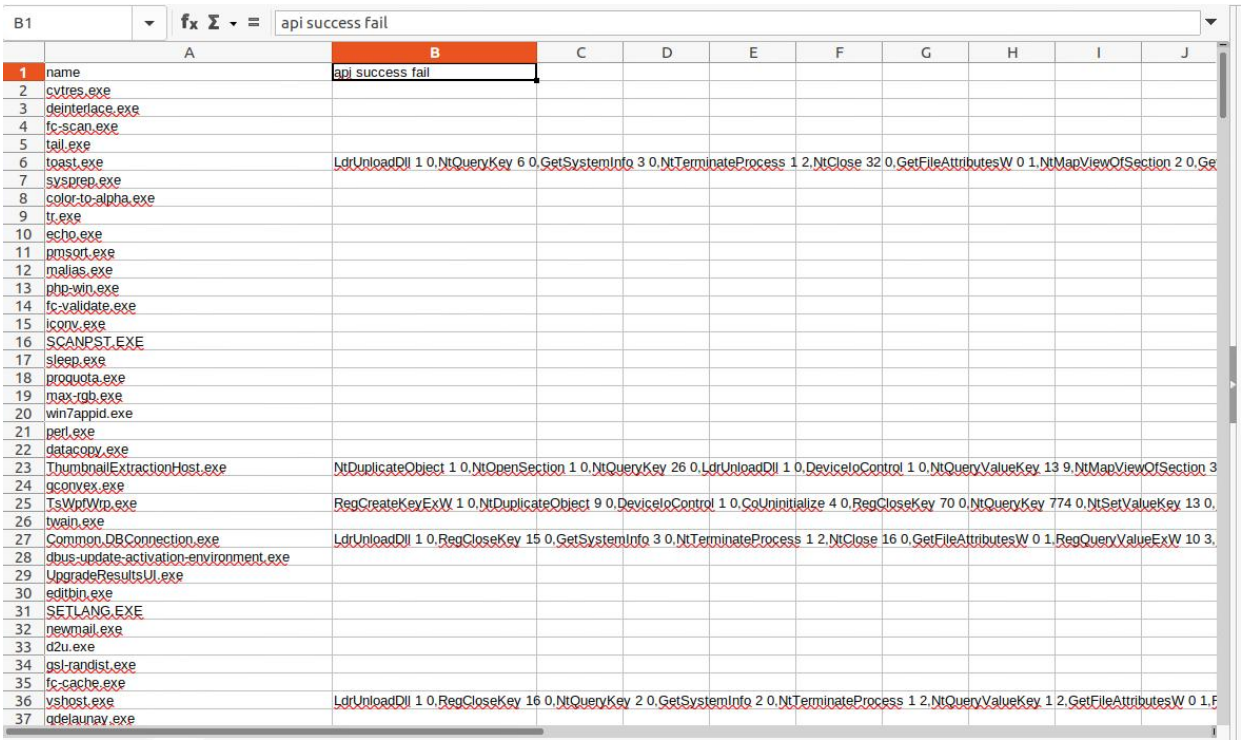
- /tasks/report: Trả về báo cáo được liên kết với ID tác vụ đã chỉ định. Thực hiện khi một mẫu có trạng thái là reported. Sau khi lấy về báo cáo sẽ thực hiện trích xuất các giá trị liên quan đến API call, Registry Key operation, File operation, DNS request, URL access.
- /tasks/delete: Xóa tác vụ đã cho khỏi cơ sở dữ liệu và xóa kết quả. Được sử dụng sau khi đã trích xuất xong các giá trị, giúp giảm tải bộ nhớ

Thực hiện phân tích động với thư mục Benign Test.



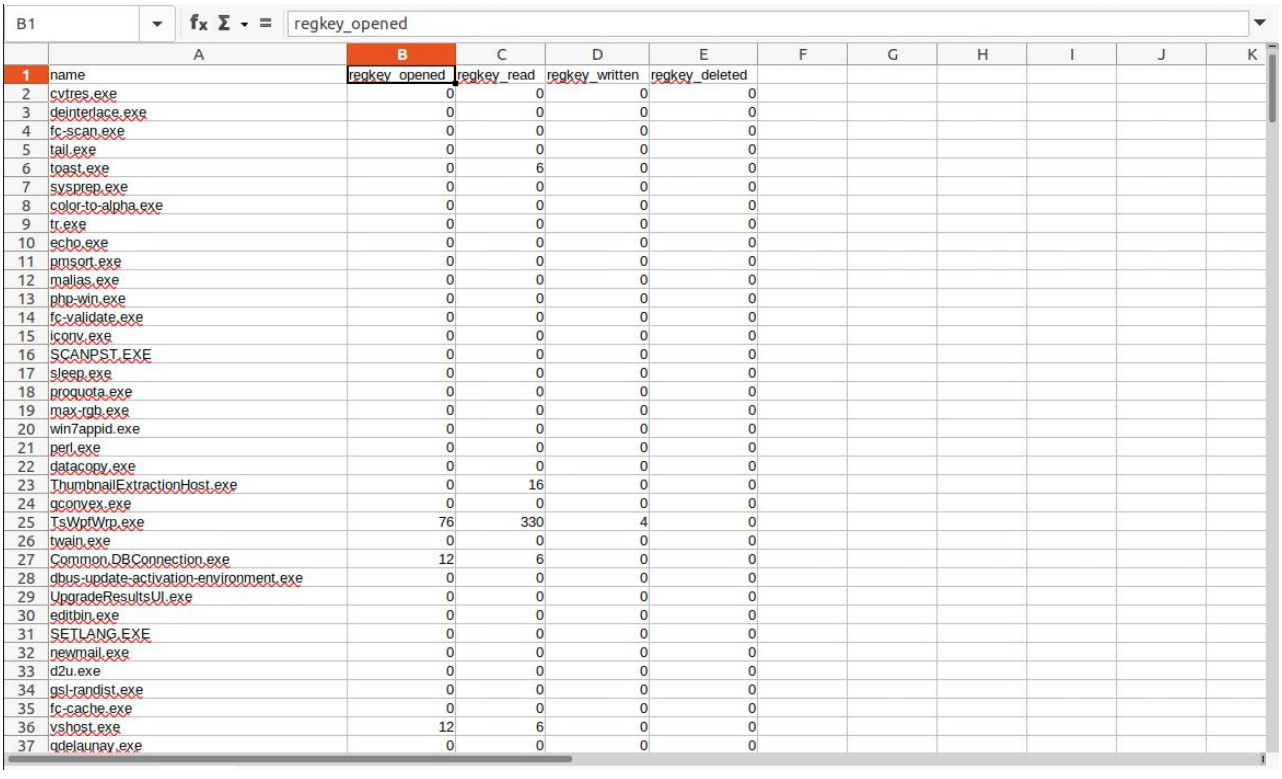
Kết quả được lưu trong các tệp apis.csv, registry.csv, files.csv, dns.csv, urls.csv.

api.csv



	A	B	C	D	E	F	G	H	I	J
1	name	api success fail								
2	cvtres.exe									
3	deinterlace.exe									
4	fc-scan.exe									
5	tail.exe									
6	toast.exe	LdrUnloadDll 1 0,NtQueryKey 6 0,GetSystemInfo 3 0,NtTerminateProcess 1 2,NtClose 32 0,GetFileAttributesW 0 1,NtMapViewOfSection 2 0,Ge								
7	sysprep.exe									
8	color-to-alpha.exe									
9	tr.exe									
10	echo.exe									
11	pmsort.exe									
12	malias.exe									
13	php-win.exe									
14	fc-validate.exe									
15	iconv.exe									
16	SCANPST.EXE									
17	sleep.exe									
18	proquota.exe									
19	max-rgb.exe									
20	win7appid.exe									
21	perl.exe									
22	datacopy.exe									
23	ThumbnailExtractionHost.exe	NtDuplicateObject 1 0,NtOpenSection 1 0,NtQueryKey 26 0,LdrUnloadDll 1 0,DeviceIoControl 1 0,NtQueryValueKey 13 9,NtMapViewOfSection 3								
24	gconvex.exe									
25	TsWpfWrp.exe	RegCreateKeyExW 1 0,NtDuplicateObject 9 0,DeviceIoControl 1 0,CoUninitialize 4 0,RegCloseKey 70 0,NtQueryKey 774 0,NtSetValueKey 13 0,								
26	twain.exe									
27	Common.DBConnection.exe	LdrUnloadDll 1 0,RegCloseKey 15 0,GetSystemInfo 3 0,NtTerminateProcess 1 2,NtClose 16 0,GetFileAttributesW 0 1,RegQueryValueExW 10 3,								
28	dbus-update-activation-environment.exe									
29	UpgradeResultsUI.exe									
30	editbin.exe									
31	SETLANG.EXE									
32	newmail.exe									
33	d2u.exe									
34	gsl-randist.exe									
35	fc-cache.exe									
36	vshost.exe	LdrUnloadDll 1 0,RegCloseKey 16 0,NtQueryKey 2 0,GetSystemInfo 2 0,NtTerminateProcess 1 2,NtQueryValueKey 1 2,GetFileAttributesW 0 1,F								
37	qdelaunay.exe									

registry.csv



	A	B	C	D	E	F	G	H	I	J	K
1	name	regkey_opened	regkey_read	regkey_written	regkey_deleted						
2	cvtres.exe	0	0	0	0						
3	deinterlace.exe	0	0	0	0						
4	fc-scan.exe	0	0	0	0						
5	tail.exe	0	0	0	0						
6	toast.exe	0	6	0	0						
7	sysprep.exe	0	0	0	0						
8	color-to-alpha.exe	0	0	0	0						
9	tr.exe	0	0	0	0						
10	echo.exe	0	0	0	0						
11	pmsort.exe	0	0	0	0						
12	malias.exe	0	0	0	0						
13	php-win.exe	0	0	0	0						
14	fc-validate.exe	0	0	0	0						
15	iconv.exe	0	0	0	0						
16	SCANPST.EXE	0	0	0	0						
17	sleep.exe	0	0	0	0						
18	proquota.exe	0	0	0	0						
19	max-rgb.exe	0	0	0	0						
20	win7appid.exe	0	0	0	0						
21	perl.exe	0	0	0	0						
22	datacopy.exe	0	0	0	0						
23	ThumbnailExtractionHost.exe	0	16	0	0						
24	gconvex.exe	0	0	0	0						
25	TsWpfWrp.exe	76	330	4	0						
26	twain.exe	0	0	0	0						
27	Common.DBConnection.exe	12	6	0	0						
28	dbus-update-activation-environment.exe	0	0	0	0						
29	UpgradeResultsUI.exe	0	0	0	0						
30	editbin.exe	0	0	0	0						
31	SETLANG.EXE	0	0	0	0						
32	newmail.exe	0	0	0	0						
33	d2u.exe	0	0	0	0						
34	gsl-randist.exe	0	0	0	0						
35	fc-cache.exe	0	0	0	0						
36	vshost.exe	12	6	0	0						
37	qdelaunay.exe	0	0	0	0						

files.csv

A1 | name

	A	B	C	D	E	F	G	H	I	J	K	L
1	name	file_opened	file_read									
2	cvtres.exe	0	0									
3	deinterlace.exe	0	0									
4	fc-scan.exe	0	0									
5	tail.exe	0	0									
6	toast.exe	1	0									
7	sysprep.exe	0	0									
8	color-to-alpha.exe	0	0									
9	tr.exe	0	0									
10	echo.exe	0	0									
11	pmsort.exe	0	0									
12	malias.exe	0	0									
13	php-win.exe	0	0									
14	fc-validate.exe	0	0									
15	iconv.exe	0	0									
16	SCANPST.EXE	0	0									
17	sleep.exe	0	0									
18	proquota.exe	0	0									
19	max-rgb.exe	0	0									
20	win7appid.exe	0	0									
21	perl.exe	0	0									
22	datacopy.exe	0	0									
23	ThumbnailExtractionHost.exe	2	0									
24	gconvex.exe	0	0									
25	TsWpfWrp.exe	32	2									
26	twain.exe	0	0									
27	Common.DBConnection.exe	1	0									
28	dbus-update-activation-environment.exe	0	0									
29	UpgradeResultsUI.exe	0	0									
30	editbin.exe	0	0									
31	SETLANG.EXE	0	0									
32	newmail.exe	0	0									
33	d2u.exe	0	0									
34	gsl-randist.exe	0	0									
35	fc-cache.exe	0	0									
36	vshost.exe	2	0									
37	qdelaunay.exe	0	0									

dns.csv

A1 | name

	A	B
1	name	dns
2	cvtres.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
3	deinterlace.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
4	fc-scan.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
5	tail.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
6	toast.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
7	sysprep.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
8	color-to-alpha.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
9	tr.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
10	echo.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
11	pmsort.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
12	malias.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
13	php-win.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
14	fc-validate.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
15	iconv.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
16	SCANPST.EXE	dns.msftncsi.com,teredo.ipv6.microsoft.com,
17	sleep.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
18	proquota.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
19	max-rgb.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
20	win7appid.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
21	perl.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
22	datacopy.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
23	ThumbnailExtractionHost.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
24	gconvex.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
25	TsWpfWrp.exe	www.msftncsi.com,crl.microsoft.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
26	twain.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
27	Common.DBConnection.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
28	dbus-update-activation-environment.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
29	UpgradeResultsUI.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
30	editbin.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
31	SETLANG.EXE	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
32	newmail.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
33	d2u.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
34	gsl-randist.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
35	fc-cache.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,
36	vshost.exe	dns.msftncsi.com,teredo.ipv6.microsoft.com,
37	qdelaunay.exe	www.msftncsi.com,dns.msftncsi.com,teredo.ipv6.microsoft.com,

CHƯƠNG 4: HƯỚNG PHÁT TRIỂN

Trong tương lai/khóa luận tốt nghiệp sắp tới, tác giả sẽ tiếp tục các nội dụng sau:

- Thực hiện tìm hiểu và trích xuất các thuộc tính mới.
- Xây dựng tập dataset hoàn chỉnh, chất lượng cao.
- Dùng phương pháp học máy, học sâu đã biết để đánh giá bộ dataset.

TÀI LIỆU THAM KHẢO

[1] Jagsir Singh, Jaswinder Singh,

A survey on machine learning-based malware detection in executable files,

Journal of Systems Architecture,

Volume 112,

2021.

[2] Rafiqul Islam, Ronghua Tian, Lynn M. Batten, Steve Versteeg,

Classification of malware based on integrated static and dynamic features,

Journal of Network and Computer Applications,

Volume 36, Issue 2,

2013.

[3] M. Ijaz, M. H. Durad and M. Ismail, "*Static and Dynamic Malware Analysis Using Machine Learning*," 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 2019.

[4] Z. Fang, J. Wang, J. Geng and X. Kan, "*Feature Selection for Malware Detection Based on Reinforcement Learning*," in IEEE Access, vol. 7, pp. 176177-176187, 2019.

[5] Damaševičius, R.; Venčkauskas, A.; Toldinas, J.; Grigaliūnas, Š. Ensemble-Based Classification Using Neural Networks and Machine Learning Models for Windows PE Malware Detection. Electronics 2021, 10, 485.

[6] Lin, C.-T & Wang, N.-J & Xiao, Han & Eckert, Claudia. (2015). Feature Selection and Extraction for Malware Classification. Journal of Information Science and Engineering 31. 965-992.