THÔNG TIN CHUNG CỦA BÁO CÁO

- Link YouTube video của báo cáo: https://youtu.be/tXH_nX5-BrU
- Link slide: https://github.com/revirven/CS2205.MAR2024/blob/main/230202027_xCS2205.DeCuong.FinalReport.Slide.pdf
- *Mỗi thành viên của nhóm điền thông tin vào một dòng theo mẫu bên dưới*
- *Sau đó điền vào Đề cương nghiên cứu (tối đa 5 trang), rồi chọn Turn in*

• Họ và Tên: Trần Nguyễn Đức Huy • MSSV: 230202027 	• Lớp: CS2205.MAR2024 • Tự đánh giá (điểm tổng kết môn): 8/10 • Số buổi vắng: 2 • Số câu hỏi QT cá nhân: 0 • Link Github: https://github.com/revirven/CS2205.MAR2024

ĐỀ CƯƠNG NGHIÊN CỨU

TÊN ĐỀ TÀI (IN HOA)

PHÁT HIỆN MÃ ĐỘC LẨN TRÁNH DỰA TRÊN OPCODE SỬ DỤNG MẠNG NƠ-RON HỌC SÂU HỒI QUY KẾT HỢP XỬ LÝ NGÔN NGỮ TỰ NHIÊN

TÊN ĐỀ TÀI TIẾNG ANH (IN HOA)

OPCODE-BASED EVASIVE MALWARE DETECTION USING DEEP RECURRENT NEURAL NETWORK AND NATURAL LANGUAGE PROCESSING

TÓM TẮT *(Tối đa 400 từ)*

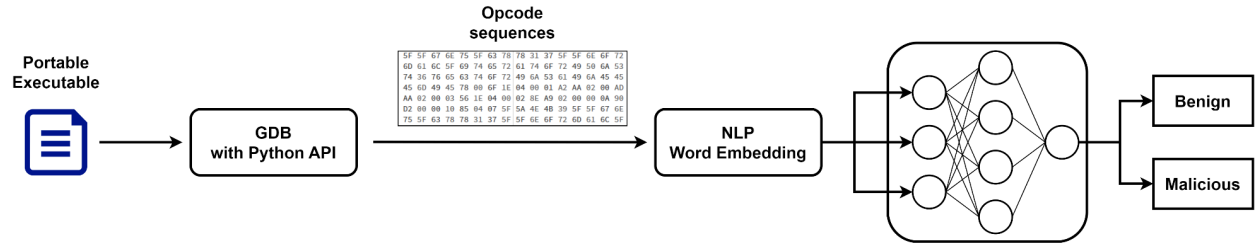
Mã độc là các phần mềm xâm nhập vào máy tính nạn nhân và thực hiện các hành vi độc hại nhằm mục đích đánh cắp thông tin cá nhân hoặc gây phương hại đến tài chính và uy tín của nạn nhân. Dưới sự phát triển của học máy, các mô hình máy học đã và đang được áp dụng trong các giải pháp phát hiện mã độc nhằm dự đoán các loại mã độc chưa từng xuất hiện. Với giả thuyết rằng việc áp dụng xử lý ngôn ngữ tự nhiên trên opcode của chương trình có thể giúp mô hình học máy đúc kết được hành vi của tệp thực thi, trong đề tài nghiên cứu này, chúng tôi xây dựng một giải pháp phát hiện mã độc lẩn tránh dựa trên opcode của tệp tin được thực thi sử dụng mạng nơ-ron học sâu hồi quy kết hợp với các phương pháp xử lý ngôn ngữ tự nhiên để nâng cao khả năng nhận diện mã độc của mô hình học máy. Opcode từ các tệp thực thi sẽ được trích xuất sử dụng trình gỡ lỗi và dịch ngược GDB thông qua Python API và được xử lý word embedding bằng các thuật toán xử lý ngôn ngữ tự nhiên trước khi được đưa vào huấn luyện mạng học sâu hồi quy.

GIỚI THIỆU *(Tối đa 1 trang A4)*

Trong kỷ nguyên số ngày nay, máy tính trở thành một phần không thể thiếu trong cuộc sống của mỗi người. Máy tính mang lại các công cụ tiện lợi hỗ trợ người dùng trong nhiều tác vụ khác nhau từ học tập, giải trí cho đến công việc. Một trong những mối đe dọa mà người dùng đang phải đối mặt khi sử dụng máy tính chính là khả năng

bị lây nhiễm bởi mã độc. Đây là những chương trình độc hại xâm nhập vào thiết bị của người dùng nhằm đánh cắp thông tin cá nhân hoặc thực hiện các hành vi phá hoại khác gây phương hại đến người dùng. Hiện nay trên thị trường có rất nhiều giải pháp phòng chống mã độc, phổ biến nhất là ở dạng các phần mềm Antivirus, tiêu biểu có thể kể đến như McAfee, Avast, Kaspersky… Hầu hết các phần mềm này đều phát hiện mã độc dựa trên đặc trưng của chúng (signature-based). Phương pháp này sử dụng một cơ sở dữ liệu các đặc trưng được thu thập từ các mẫu mã độc đã được tìm thấy để so sánh và xác định liệu một tệp tin có phải là độc hại hay không. Phương pháp này thiếu hiệu quả trước các mẫu mã độc sử dụng kỹ thuật rối mã (obfuscation) và không thể phát hiện được các loại mã độc mới không có trong cơ sở dữ liệu. Để khắc phục những yếu điểm này, các giải pháp phát hiện mã độc áp dụng máy học với nhiều hướng tiếp cận khác nhau đã được đề xuất nhằm dự đoán các loại mã độc chưa từng xuất hiện. Giải pháp [4] của Li Yang và cộng sự sử dụng kết hợp giữa các thuật toán Gradient Boosting và mạng nơ-ron tích chập để phát hiện mã độc dựa trên nhiều đặc trưng khác nhau như raw bytes, byte codes, PE Imports và PE Section names. Giải pháp này đạt độ chính xác cao trên tập dữ liệu Malshare nhưng bị hạn chế bởi tốc độ huấn luyện mô hình do sử dụng 2 mô hình khác nhau trong quá trình huấn luyện. [1] là một hướng tiếp cận khác được đề xuất bởi Mohamoud Kalash và cộng sự, cũng sử dụng mạng nơ-ron tích chập nhưng dựa trên mã nhị phân của tệp thực thi được chuyển đổi thành ma trận 2 chiều, biểu diễn dưới dạng ảnh greyscale. Hướng tiếp cận này khá phổ biến nhưng kém hiệu quả trước những loại mã độc được áp dụng kỹ thuật làm rối mã. Giải pháp [3] của Deniz Demirci và cộng sự so sánh mức độ hiệu quả giữa việc sử dụng mô hình xử lý ngôn ngữ tự nhiên và mạng nơ-ron học sâu hồi quy trong bài toán phân loại mã độc sử dụng opcode ở dạng plain text, cho thấy khả năng của mô hình xử lý ngôn ngữ tự nhiên khi được áp dụng trong ngữ cảnh của bài toán phân loại mã độc. Điểm hạn chế chung của các nghiên cứu trên là giải pháp chỉ có thể phát hiện được những loại mã độc đơn giản không được áp dụng các kỹ thuật lẩn tránh do chỉ huấn luyện mô hình thông qua phân tích tĩnh. Với giả thuyết rằng việc áp dụng xử lý ngôn ngữ tự nhiên trên opcode của chương trình có thể giúp mô

hình học máy đúc kết được hành vi của tệp thực thi, trong đề tài nghiên cứu này, chúng tôi xây dựng một giải pháp phát hiện mã độc ở thời gian thực dựa trên opcode sử dụng mạng nơ-ron hồi quy kết với các kỹ thuật xử lý ngôn ngữ tự nhiên giúp nâng cao khả năng nhận diện mã độc của mô hình học máy.



Khi được cung cấp một tệp thực thi, giải pháp đề xuất sẽ thực hiện debug, trích xuất opcode thông qua GDB. Opcode thu thập được sẽ được phân tích thông qua các kỹ thuật xử lý ngôn ngữ tự nhiên, tạo nên các vector đặc trưng mới đại diện cho các opcode thu thập được. Các vector này sẽ được đưa vào mạng nơ-ron học sâu hồi quy để huấn luận mô hình phát hiện mã độc. Với lớp đầu ra 1 nút, mạng học sâu cho ra kết quả là một trong hai giá trị 0 và 1 xác định liệu tệp thực thi có phải là độc hại hay không.

MỤC TIÊU

(Viết trong vòng 3 mục tiêu, lưu ý về tính khả thi và có thể đánh giá được)

- Thu thập và xây dựng bộ dữ liệu mã độc
- Xây dựng huấn luyện mô hình phát hiện mã độc
- Áp dụng và đánh giá mức độ ảnh hưởng của các kỹ thuật xử lý ngôn ngữ tự nhiên trong việc phân tích opcode của tệp thực thi

NỘI DUNG VÀ PHƯƠNG PHÁP

(Viết nội dung và phương pháp thực hiện để đạt được các mục tiêu đã nêu)

NỘI DUNG

- **Dataset:** Vì giải pháp được đề xuất sử dụng opcode làm đặc trưng để phát hiện

mã độc trong thời gian thực, tập dữ liệu được sử dụng cần phải ở dạng các tệp PE vẫn có khả năng thực thi.

- **Trích xuất đặc trưng:** Opcode từ các tệp PE có thể được trích xuất thông qua trình dịch ngược GDB. Sử dụng Python API của GDB, tệp PE có thể được thực thi và opcode có thể được thu thập trong thời gian thực.
- **Tiền xử lý dữ liệu:** Các tập dữ liệu thường không cân bằng về số lượng mẫu độc hại và mẫu lành tính, chính vì vậy ta cần sinh thêm, thu thập thêm hoặc loại bỏ bớt các mẫu dữ liệu để cân bằng tập dữ liệu.
- **Word embedding:** Opcode trước khi được đưa vào mạng nơ-ron học sâu sẽ được phân tích sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để tạo nên các đầu vào có ý nghĩa giúp mô hình học sâu đúc kết được hành vi của tệp thực thi.
- **Huấn luyện mạng nơ-ron học sâu hồi quy:** Opcode sau khi được xử lý sẽ được sử dụng làm đầu vào của mạng nơ-ron học sau hồi quy để huấn luyện mô hình phát hiện mã độc.
- Đánh giá mô hình huấn luyện thông qua các thước đo Accuracy, Precision, Recall

PHƯƠNG PHÁP

- Tìm hiểu và thu thập các tập dữ liệu mã độc đang được sử dụng phổ biến trong các nghiên cứu hiện tại như BODMAS, EMBER, SoReL-20M,…
- Tìm hiểu và áp dụng các mô hình xử lý ngôn ngữ tự nhiên để phân tích opcode như Word2Vec, GloVe, BERT, GPT,…
- Tìm hiểu về GDB và các Python API của GDB có thể giúp thu thập opcode trong quá trình debug một tệp thực thi. Viết chương trình thực hiện tác vụ này
- Tìm hiểu các mô hình mạng nơ-ron học sâu hồi quy như RNN, LSTM, GRU,… và xây dựng một một mô hình phù hợp với giải pháp hiện tại
- Xây dựng một mô hình riêng biệt không được áp dụng xử lý ngôn ngữ tự nhiên để đánh giá mức độ ảnh hưởng của phương pháp này

KẾT QUẢ MONG ĐỢI

(Viết kết quả phù hợp với mục tiêu đặt ra, trên cơ sở nội dung nghiên cứu ở trên)

- Giải pháp đề xuất có khả năng phát hiện được các họ mã độc đa hình hay các họ mã độc được áp dụng các kỹ thuật lẩn tránh khác
- Mô hình xử lý ngôn ngữ tự nhiên có tác động tích cực đến độ chính xác của mô hình học sâu được xây dựng

TÀI LIỆU THAM KHẢO *(Định dạng DBLP)*

[1] Mahmoud Kalash, Mrigank Rochan, Noman Mohammed, Neil D. B. Bruce, Yang Wang, Farkhund Iqbal: Malware Classification with Deep Convolutional Neural Networks. NTMS 2018: 1-5

[2] Enes Sinan Parildi, Dimitrios Hatzinakos, Yuri A. Lawryshyn: Deep learning-aided runtime opcode-based Windows malware detection. Neural Comput. Appl. 33(18): 11963-11983 (2021)

[3] Deniz Demirci, Nazenin Sahin, Melih Sirlanci, Cengiz Acartürk: Static Malware Detection Using Stacked BiLSTM and GPT-2. IEEE Access 10: 58488-58502 (2022)

[4] Li Yang, Junlin Liu:TuningMalconv: Malware Detection With Not Just Raw Bytes. IEEE Access 8: 140915-140922 (2020)