Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: Cơ chế hoạt động của mã độc

Kỳ báo cáo: Đồ án thực hành

Tên chủ đề: Hiện thực framework FUMVar-Ex

GV: Nghi Hoàng Khoa

Ngày báo cáo: 27/5/2023

Nhóm: 1

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Nguyễn Bùi Kim Ngân	20520648	20520648@gm.uit.edu.vn
2	Nguyễn Bình Thục Trâm	20520815	20520815@gm.uit.edu.vn
3	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Tóm tắt mục tiêu đề tài và những đóng góp của bài báo	100%	
2	Phương pháp thực hiện gồm thành phần hệ thống và các loại perturbations	100%	
3	Kết quả thực nghiệm của nhóm tác giả	100%	
4	Kết quả hiện thực framework FUMVar-Ex của nhóm	100%	

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

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

TÀI LIỆU TRÌNH BÀY HIỆN THỰC

Tóm tắt sơ lược về các phần của tài liệu này

1) Tóm tắt mục tiêu đề tài và những đóng góp của bài báo
2) Phương pháp thực hiện gồm thành phần hệ thống và các loại perturbations
3) Kết quả thực nghiệm của nhóm tác giả
4) Kết quả hiện thực framework FUMVar-Ex của nhóm

1. Mục tiêu đề tài

Mục tiêu tạo ra biến thể malware trên file PE bằng cách dùng các kỹ thuật trốn tránh để bypass malware detectors cụ thể là các phần mềm thương mại anti-malware.

Các đóng góp chính của bài báo:

- Nhóm tác giả trình bày framework FUMVar-Ex, được cải tiến từ framework trước đó là FUMVar, có nhiệm vụ tạo ra biến thể malware với các kỹ thuật trốn tránh cải tiến để bypass malware detectors.
- Giới thiệu phương pháp gây nhiễu loạn mới và phức tạp giúp cải thiện đáng kể khả năng lẩn tránh của các biến thể malware.
- Đề xuất phương pháp đánh giá tổng hợp về xem xét các hành vi thay đổi theo runtime để cải thiện độ hiệu lực của việc kiểm tra độ giống nhau về hành vi giữa biến thể malware được tạo với nguyên mẫu.
- So sánh hiệu suất với các framework hiện nay về chủ đề tạo biến thể malware.
- Thử nghiệm thực tế framework lên các phần mềm thương mại anti-malware.
- Phân tích độ hiệu quả của các phương pháp gây nhiễu loạn, các loại malware và kiểm tra đặc điểm của các biến thể malware lẩn tránh. Từ đó, tăng cường và cải thiện các giải pháp cho anti-malware.

2. Phương pháp thực hiện

Thứ nhất, về thành phần hệ thống được đề xuất gồm 5 modules sau:

Parser module - Trích xuất các sections từ file PE

Input: Mẫu malware dưới dạng PE file

Nhiệm vụ: Kiểm tra những sections và field của file PE có thể bị làm rối mà không gây tổn tại tới tính hiệu lực và chức năng của file.

Output: Các sections có thể bị xáo trộn và gắn thẻ cho các field

Modifier module - Áp dụng đột biến lên file PE

Input: Output của parser module

Nhiệm vụ: Thực hiện biến đổi một mẫu malware bằng cách dùng Genetic Algorithm (GA) chọn ngẫu nhiên một nhiễu loạn (perturbations) vào mỗi lần đột biến và gắn thêm vào byte code.

Output: Các biến thể malware

Verifier module - Kiểm tra các biến thể bị lỗi

Input: Output của modifier module

Nhiệm vụ: Kiểm tra biến thể nào khiến file PE bị lỗi bằng cách dùng Cuckoo sandbox và lọc bỏ chúng.

Output: các biến thể không bị lỗi

Validator module - Kiểm tra hành vi của malware

Input: Output của verifier module

Nhiệm vụ: Cần đảm bảo rằng các hành vi của biến thể malware được tạo giống với mẫu malware nguyên gốc và dùng Cuckoo sandbox để kiểm tra dựa trên các signatures

Output: Các biến thể có hành vi giống với nguyên mẫu

Scorer module - Đo lường các biến thể malware

Input: Output của Validator module

Nhiệm vụ: Đo lường khả năng trốn tránh của các biến thể malware đã tạo bằng cách dùng VirusTotal nơi chứa 71 anti-malware solutions. Cuối cùng tính tổng điểm cho biến thể dựa trên kết quả của VirusTotal (số lượng anti-malware solutions bypass thành công) và sự giống nhau giữa biến thể được tạo với mẫu malware nguyên bản.

Output: Điểm của mỗi biến thể malware

Thứ hai, về các loại nhiễu loạn (perturbations) được dùng gồm 14 loại chia làm 2 kiểu sau:

Non-behavioral changes (NBC)

1) Overlay Append (OA)
2) Dos Header (DH)
3) Dos Stub (DS)
4) Coff Header (CH)
5) Optional Header (OH)
6) Data Directory (DD)
7) Rich Header (RH)

Behavioral changes (BC) có thể ảnh hưởng tới chức năng của chương trình

1) Section Rename (SRN)
2) Section Add (SAD)
3) Section Append (SAP)
4) Code Cave Inject (CI)
5) Pack (PA)
6) Unpack (UP)
7) Xor Obfuscation (XO)

3. Kết quả thực nghiệm

Nhóm tác giả đã thực nghiệm và kiểm tra độ hiệu quả của framework như sau:

FUMVar-Ex được so sánh với một số phương pháp tạo biến thể malware hiện có là: FUMVar, AIMED, RL và MAB-MALWARE dựa trên 2 chỉ số là tỉ lệ tạo thành công biến thể hợp lệ và tỉ lệ lẩn tránh.

FUMVar-Ex được thử nghiệm lên các phần mềm thương mại anti-malware là Avast, AVG, BitDefender, Kaspersky, Malwarebytes, McAfee và TrendMicro, đưa ra số liệu về tỉ lệ thành công lẩn tránh khỏi phát hiện giữa các phần mềm kể trên của FUMVar-Ex cùng với một số phương pháp tạo biến thể malware đã nói.

Đánh giá độ hiệu quả của mỗi perturbation dựa trên:

- Tỉ lệ lẩn tránh phát hiện giữa các phần mềm thương mại anti-malware của 14 loại perturbation
- Tỉ lệ tạo thành công biến thể hợp lệ của 14 loại perturbation
- Tỉ lệ lẩn tránh của 14 loại perturbation

4. Demo của nhóm

Cấu hình máy:

CPU: Intel Core i5 8250U (2 nhân 2 luồng)

Ram: 4GB DDR3L

HDD: 80GB

Card mạng: NAT

OS: Ubuntu 20.04.5

Yêu cầu env:

Python 3.8.10

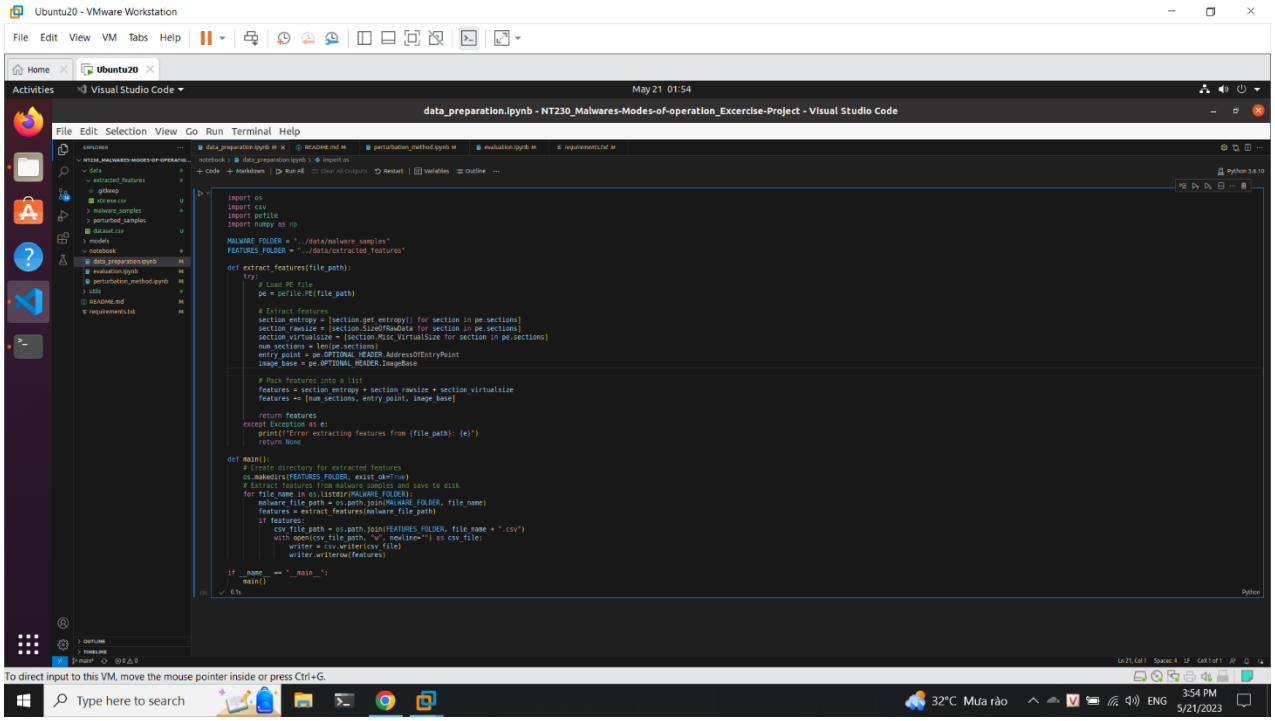
Các thư viện python cho demo:

- numpy==1.19.5
- pandas==1.2.1
- scikit-learn==0.24.1
- tensorflow==2.4.0
- matplotlib==3.3.4
- seaborn==0.11.1

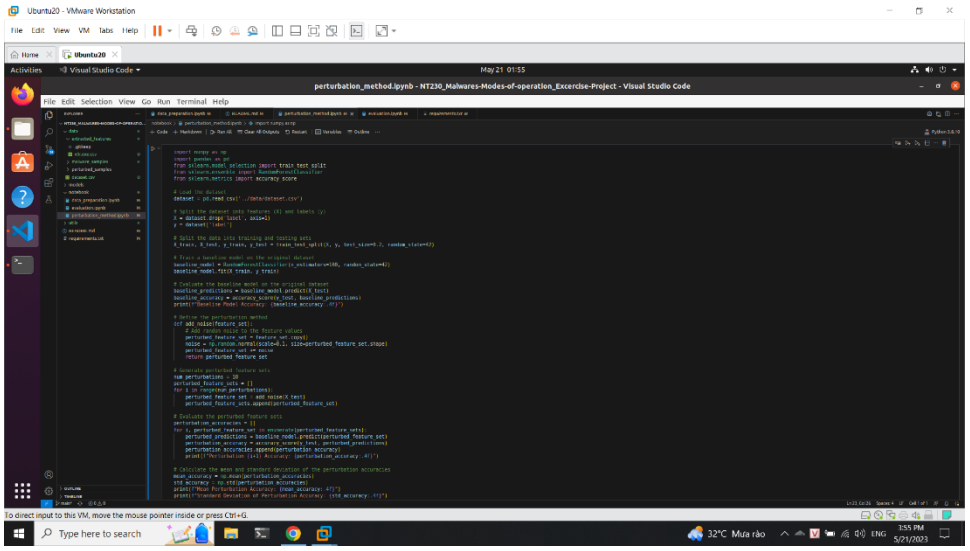
IDE: VS Code

Code chính:

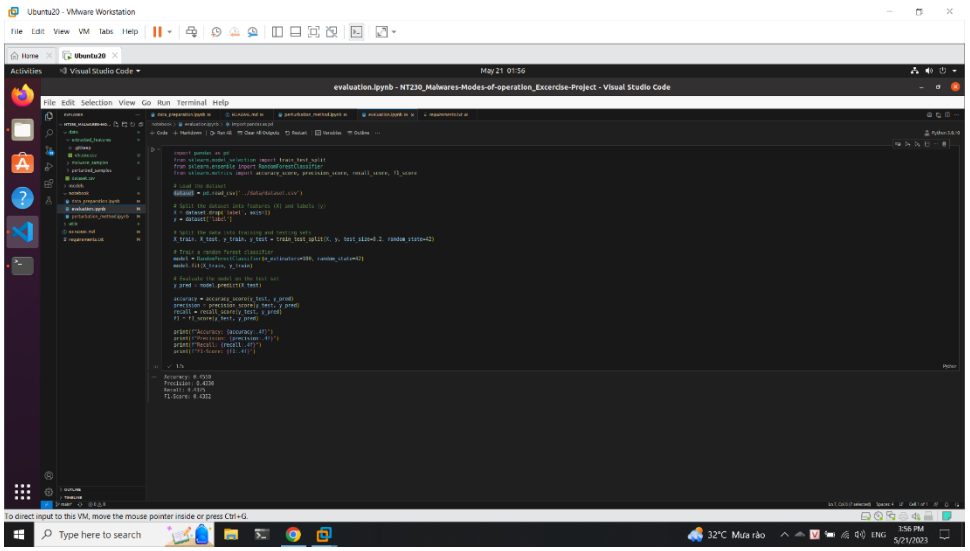
Pha 1: Xử lý dữ liệu



Pha 2: Perturbation

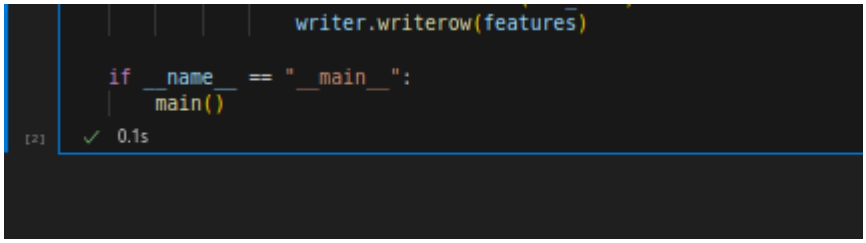


Pha 3: Evaluation

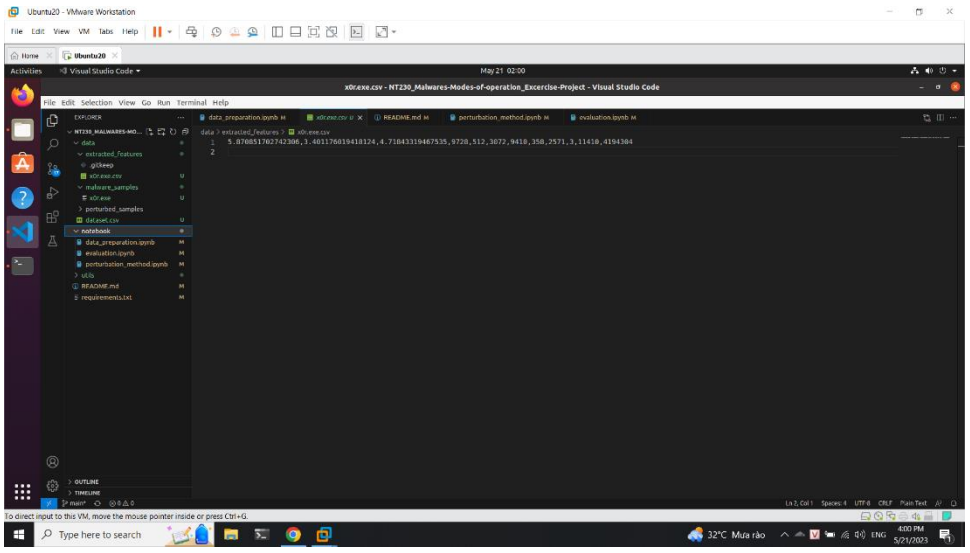


Thực hiện chạy:

1. Run `notebooks/data_preparation.ipynb` to preprocess the malware dataset.

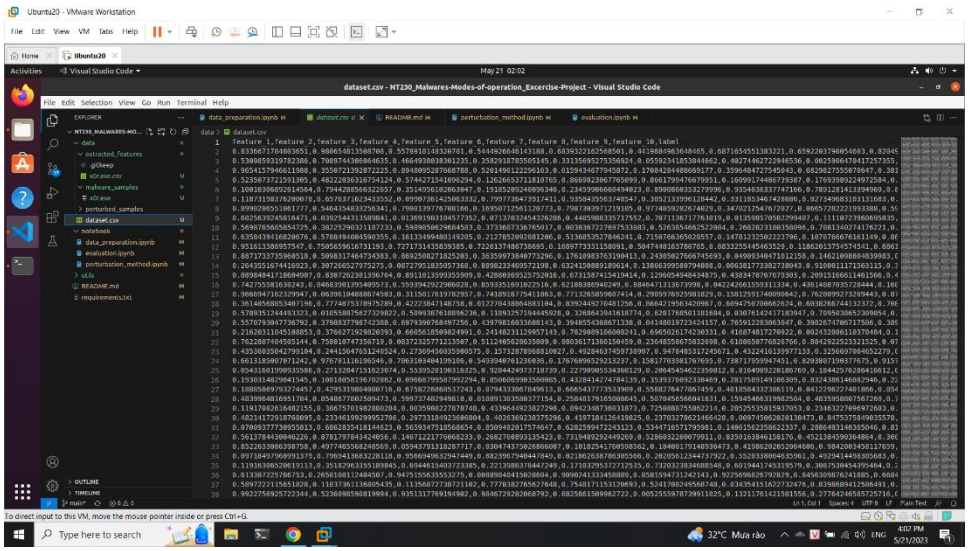


Thực hiện với malware xor



Thực hiện tải các file malware từ: https://github.com/ytisf/theZoo

Và thực hiện tạo dataset



2. Run `notebooks/perturbation_method.ipynb` to generate perturbed samples of the malware dataset.

```
[1]  ✓ 2.2s

...  Baseline Model Accuracy: 0.4550
     Perturbation 1 Accuracy: 0.4450
     Perturbation 2 Accuracy: 0.4650
     Perturbation 3 Accuracy: 0.5150
     Perturbation 4 Accuracy: 0.4800
     Perturbation 5 Accuracy: 0.4800
     Perturbation 6 Accuracy: 0.4950
     Perturbation 7 Accuracy: 0.4450
     Perturbation 8 Accuracy: 0.4900
     Perturbation 9 Accuracy: 0.4950
     Perturbation 10 Accuracy: 0.4450
     Mean Perturbation Accuracy: 0.4755
     Standard Deviation of Perturbation Accuracy: 0.0234
```

Ta thấy được độ accuracy của từng pertuabation

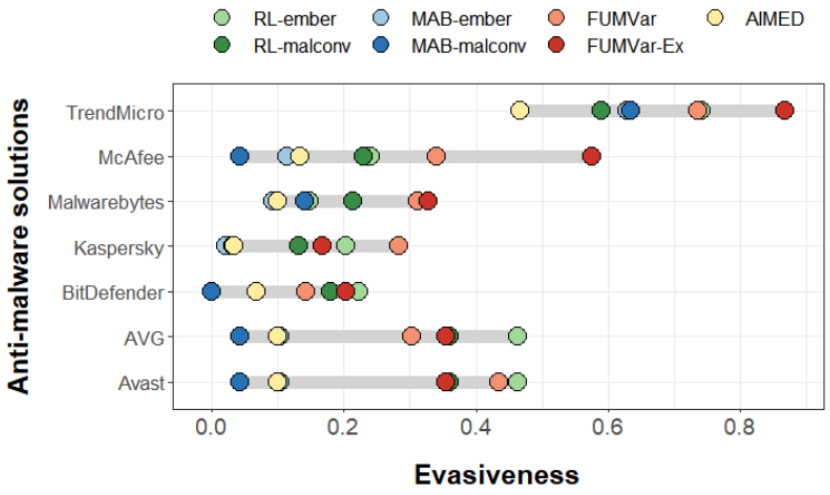
Ở dưới ta có độ nhiễu trung bình 47.55% và độ lệch chuẩn của nhiễu là 0.0234

3. Run `notebooks/evaluation.ipynb` to train and evaluate a machine learning model on the original and perturbed datasets.

```
[1]  ✓ 1.7s

...  Accuracy: 0.4550
     Precision: 0.4330
     Recall: 0.4375
     F1-Score: 0.4352
```

Cuối cùng khi ta chạy xong pha 2 thì ta sẽ sử dụng thử viện sklearn.metrics để tính điểm như hình và đánh giá mức điểm này đang giống với kết quả của AVG và Avast như bài báo



Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

  *Ví dụ: [*NT101.K11.ANTT*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT