BÁO CÁO TỔNG KẾT ĐỒ ÁN MÔN HỌC

**Môn học: Cơ chế hoạt động của mã độc**

**Tên chủ đề: File Packing in Malware: Creation and Detection**

*Mã nhóm:* ***G1****. Mã đề tài:* ***S02***

**Lớp**: **NT230.N21.ATCL**

1. **THÔNG TIN THÀNH VIÊN NHÓM:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Mạnh Cường | 20520421 | 20520421@gm.uit.edu.vn |
| 2 | Hoàng Văn Anh Đức | 20520890 | 20520890@gm.uit.edu.vn |
| 3 | Lê Quang Minh | 20520245 | 20520245@gm.uit.edu.vn |

1. **TÓM TẮT NỘI DUNG THỰC HIỆN:**
   1. **Chủ đề nghiên cứu trong lĩnh vực Mã độc:**

🗹 Phát hiện mã độc

* 1. **Liên kết lưu trữ mã nguồn của nhóm:**

Mã nguồn của đề tài đồ án được lưu tại: <https://github.com/hiimicebear/Packed_Malware>

* 1. **Tên bài báo tham khảo chính:**

|  |
| --- |
| Aghakhani, H., Gritti, F., Mecca, F., Lindorfer, M., Ortolani, S., Balzarotti, D., ... & Kruegel, C. (2020, January). When malware is packin'heat; limits of machine learning classifiers based on static analysis features. In Network and Distributed Systems Security (NDSS) Symposium 2020. |

* 1. **Dịch tên Tiếng Việt cho bài báo:**

|  |
| --- |
| Khi đóng gói mã độc là xu hướng; hạn chế của các mô hình học máy phân loại dựa trên các thuộc tính phân tích tĩnh. |

* 1. **Tóm tắt nội dung chính:**

|  |
| --- |
| Bài báo nghiên cứu về những hạn chế của các mô hình học máy phân loại dựa trên phân tích tĩnh khi đối mặt với những mẫu mã độc được đóng gói. Bài báo nghiên cứu về những vấn đề, thí nghiệm chứng minh cho vấn đề và đưa ra hướng giải quyết tương đối khái quát, khác với các bài báo đề xuất phương pháp mới cụ thể.  Hiện nay, học máy phân loại dựa trên phân tính tĩnh những tệp thực thi mã độc được ứng dụng phổ biến, nhờ vào sự hiệu quả khi cân bằng giữa hiệu suất và tính bảo mật so với phân tích động. Tuy nhiên, nó cũng gặp phải một số vấn đề, đặc biệt kể đến những mẫu mã độc được đóng gói nhằm tránh né các trình phân loại.  Đóng gói (packing) được thực hiện bới các trình đóng gói (packer) là phương pháp áp dụng các thuật toán nén và/hoặc mã hóa vào tệp thực thi nhằm ẩn nội dụng và giảm dụng lượng. Việc đóng gói vẫn đảm bảo được chức năng gốc của tệp thực thi đó, nhờ vào các trình giải đóng gói (unpacker) khi tệp được chạy. Do đó, các mô hình phân loại tĩnh khó phát hiện được tính chất độc hại của tệp, khi bản chất mô hình sẽ không chạy thực thi nó.  Tận dụng điều đó, vào khoảng thời gian trước đây, hầu hết các tệp thực thi được đóng gói trong thực tế đều mang nhãn độc hại. Tuy nhiên, thống kê gần đây cho thấy, số lượng phần mềm lành tính áp dụng đóng gói nhằm bảo vệ tài nguyên mã nguồn cho tệp thực thi đã tăng lên đáng kể. Sự thay đổi về tỉ lệ chênh lệch giữa các nhãn nói trên trong thực tế đã đặt ra một số nghi vấn về tính hiệu quả của các giải pháp được cho là có thể chống lại mã độc đóng gói được công bố trước đó.  Vì vậy, tác giả thực hiện một số thí nghiệm kiểm chứng vấn đề trên, và tạo ra 2 datasets nhằm hỗ trợ thí nghiệm và là giải pháp huấn luyện cải thiện các mô hình học máy phân loại để chống lại các mẫu đóng gói. |

* 1. **Tóm tắt các kỹ thuật chính được mô tả sử dụng trong bài báo:**

|  |
| --- |
| Như đã nhắc đến ở trên, tác giả sẽ tạo ra 2 datasets để hổ trợ cho các thí nghiệm chứng minh của mình. Đồng thời, với việc đảm bảo độ chính xác cho các nhãn, trích xuất thuộc tính giá trị và chọn lọc tỉ lệ hợp lý, các datasets sẽ là giải pháp cho việc huấn luyện các mô hình học máy phân loại đạt hiệu quả khi đối mặt với các mẫu được đóng gói. Tất cả các mẫu thực thi trong 2 tập dữ liệu đều thuộc định dạng PE trên hệ điều hành Windows x86.   1. **Wild dataset:** được gọi tên là “hoang dã” vì tập dữ liệu được lấy từ các mẫu tệp thực thi tồn tại trong thực tế.  * Cụ thể, những mẫu được lấy từ: (i) các phần mềm anti-malware thương mại, (ii) EMBER dataset, tập dữ liệu được công bố trong một bài báo trước đây và được sử dụng phổ biến trong nhiều nghiên cứu về sau. * Tiếp theo, tất cả được gán nhãn malware/benign và packed/unpacked từ những nguồn có độ chính xác cao. Cụ thể, nhãn malware/benign được kiểm tra lại bằng những công cụ có độ tin cậy cao trên Virus Total, sau đó chọn lọc những mẫu có sự thống nhất về nhãn giữa kết quả từ Virus Total và nguồn của mẫu. Nhãn packed/unpacked được gán bằng các phương pháp phân tích động, nhờ vào dấu hiệu giải đóng gói khi chạy tệp, điều đó tăng độ chính xác cho nhãn. Trong đó bao gồm: (i) sandbox từ các anti-malware thương mại, (ii) Deep Packer Inspector, (iii) các công cụ khác (Manalyze, Exeinfo PE, yara rules, PEiD, F-Prot). * Cuối cùng, cho ra 50,724 mẫu, bao gồm 4,396 unpacked benign, 12,647 packed benign và 33,681 packed malicious. (Các mẫu có nhãn packed malicious bị tác giả loại bỏ do số lượng ít và không cần thiết cho thí nghiệm)      1. **Lab dataset:** tập dữ liệu là tất cả những mẫu từ Wild, nhưng được lần lượt đóng gói lại bằng 9 packers cụ thể, sau đó chọn lọc và loại bỏ một số mẫu không thể xử lý.  * 9 packers bao gồm: Obsidium, PELock, Themida, PECompact, Petite, UPX, kkrunchy, MPRESS, tElock. * Cuối cùng, cho ra 341,444 mẫu đã đóng gói (với 94.56% mẫu có thể bảo toàn được chức năng).     Ngoài ra, cả 2 datasets trên đều được trích xuất đồng bộ chung một bộ thuộc tính, bao gồm 9 nhóm với tổng cộng 56,543 thuộc tính. |

* 1. **Môi trường thực nghiệm của bài báo:**

|  |
| --- |
| Trong bài báo, tác giả đã thực hiện **12 thí nghiệm** (chia làm **4 vấn đề** khác nhau) để kiểm chứng về hiệu quả của các mô hình học máy phân loại chống lại các mẫu được đóng gói.  Tác giả không đề cập đến các thông số phần cứng và môi trường thí nghiệm, tuy nhiên, dựa theo mã nguồn tác giả công bố, các thí nghiệm được thực hiện trên hệ điều hành Linux với ngôn ngữ Python. Bên cạnh đó, đây là một số thiết lập chung cho các thí nghiệm:   * Datasets sử dụng như đã đề cập ở **F**. Thuật toán phân loại cho tất cả các thí nghiệm được thống nhất là Random Forest. * Training và testing được chia theo tỉ lệ 70%-30%. Tỉ lệ nhãn benign-malware trong cả tập train và test chia đều 50-50. * Tác giả lặp lại mỗi thí nghiệm 5 lần. Với mỗi lần, tập train-test được chọn ngẫu nhiên khác nhau. Kết quả là trung bình cộng của 5 lần thực hiện. * Tác giả dùng toàn bộ 56,543 thuộc tính (đã đề cập ở **F**) cho tất cả các thí nghiệm.   Phần **H** sẽ đề cập tóm lược nội dung từng thí nghiệm và kết quả. |

* 1. **Kết quả thực nghiệm của bài báo:**

|  |
| --- |
| Đây là tóm lược nội dung 12 thí nghiệm thuộc 4 vấn đề và kết quả trong bài báo: *(nếu không có đề cập đặc biệt, hầu hết mô hình phân loại giữa malware/benign)*   1. **Vấn đề 1:** Mô hình phân loại có nhầm lẫn một quá trình đóng gói từ một Packer cụ thể là nhãn Độc hại, khi tỉ lệ dữ liệu để huấn luyện mô hình không tối ưu?  * Thí nghiệm 1: Chỉ training với 2 nhãn Packed Malware – Unpacked Benign với Wild dataset.   Kết quả: *mô hình nhầm lẫn đặc điểm packed là malware, xem tất cả tệp được đóng gói là độc hại.*   * Thí nghiệm 2: Training mô hình phân loại các packers khác nhau với Lab dataset.   Kết quả: *tỉ lệ đánh giá Accuracy 99.99%, cho thấy việc phân loại các đặc điểm của trình đóng gói là một công việc đơn giản với các mô hình học máy.*   * Thí nghiệm 3: Với Lab dataset, training với chỉ benign từ 4 packers và chỉ malware từ 5 packers còn lại.   Kết quả: *mô hình nhầm lẫn tất cả các tệp được đóng gói từ 4 packers đều là benign và tất cả các tệp được đóng gói từ 5 packers đều là malware (tập test tỉ lệ đồng đều). Cho thấy mô hình học máy phân loại dễ nhầm lẫn một quá trình đóng gói là benign/malware bất kể tính độc hại thật sự của tệp.*   * **Kết luận, giả thuyết từ vấn đề 1 là Chính xác.**  1. **Vấn đề 2:** Vậy mô hình phát hiện mã độc tĩnh có thực sự phân loại được những mẫu đã đóng gói?  * Thí nghiệm 4, 5: Để giải quyết những hạn chế ở vấn đề 1, tác giả thêm nhãn Packed Benign vào tập training với tỉ lệ tăng dần so với tổng các Benign (tức là tăng Packed Benign – giảm Unpacked Benign, vẫn đảm bảo 50-50 cho Benign-Malware) và thực hiện loạt các thí nghiệm với cả 2 datasets. Ở Lab dataset, vì không có Unpacked Benign trong tập training, tác giả mượn các tệp đó từ tập Wild (tập testing vẫn thuộc tập Lab).   Kết quả: (i) *Wild dataset, duy trì được FP và FN ở mức thấp khi tỉ lệ Packed Benign ~50%,* (ii) *Lab dataset, vì môi trường testing đều là Packed nên FP và FN duy trì được mức thấp khi tỉ lệ Packed Benign ~100% (tức là không cần có Unpacked Benign trong training). Cho thấy, trong môi trường chỉ toàn các tệp được đóng gói, mô hình có thể xác định được tính chất độc hại của tệp. Điều đó rút ra được mặc dù bị đóng gói, các thuộc tính trích xuất tĩnh vẫn hữu ích trong phân loại benign/malware.*     * Thí nghiệm 6: Đây là thí nghiệm để xác định nhóm thuộc tính nào có giá trị cao trong phân loại, bằng cách thí nghiệm với từng nhóm thuộc tính và từng datasets.   Kết quả: *nhóm thuộc tính Bytes n-grams và PE sections có giá trị ảnh hưởng cao đến kết quả phân loại, đồng thời nêu ra tầm quan trọng của những nhóm khác là cần thiết trong quá trình phân loại.*   * **Kết luận, những thuộc tính trích xuất tĩnh vẫn hữu ích trong quá trình phát hiện tính độc hại của tệp thực thi.**  1. **Vấn đề 3:** Vậy mô hình có thể phân loại hiệu quả với những mẫu được đóng gói từ các packer chưa từng biết, và một số vấn đề khác ở môi trường thực tế không?  * Thí nghiệm 7, 8, 9: Ở những thí nghiệm thuộc vấn đề 2, tác giả chia đều tệp của các packers trong tập training và testing (tức là chỉ test với những quá trình đóng gói từ packers được biết trước trong train). Loạt thí nghiệm này sẽ training mô hình với những mẫu từ một số packers và testing chúng với những mẫu từ các packers khác. Đồng thời, thí nghiệm cũng kiểm chứng xem các mô hình có phát hiện ra được những đặc điểm đóng gói từ những packer đã đóng gói tệp ở lớp trước đó hay không (tức là một tệp có thể được đóng gói nhiều lần/lớp), bằng cách thay đổi tập train-test từ 2 datasets (tập Lab là tập Wild đã được đóng gói).   Kết quả: *mô hình không thể phân loại đúng được đối với những mẫu đã đóng gói bởi packers chưa từng thấy khi training (kể cả những đặc điểm từ packers ở lớp đóng gói trước/sau).*   * Thí nghiệm 10, 11: Thí nghiệm với những mầu đối kháng (Adversarial Samples) và những mẫu từ những packers mạnh (có những thuật toán mã hóa mạnh). Thí nghiệm ngoài đề tài nên sẽ chỉ đề cập khái quát.   Kết quả: *không hiệu quả với những mẫu đối kháng và và những mẫu từ những packers mạnh.*   * **Kết luận, giả thuyết từ vấn đề 3 là Chưa hiệu quả.**  1. **Vấn đề 4:** Những phần mềm phát hiện mã độc thương mại trong thực tế dựa trên học máy phân tích tĩnh có chống lại được các mẫu đóng gói không?  * Thí nghiệm 12: Đưa 6,000 malwares và 6,000 benigns thuộc Lab dataset lên Virus Total để kiểm tra 6 engines phát hiện mã độc (ở đây kiểm nghiệm những engines khác với những engines đã thực hiện gán nhãn cho Wild dataset).   Kết quả: *tỉ lệ FP và FN khá cao.*     * **Kết luận, một số engines phát hiện mã độc thuộc Virus Total nhầm lẫn đặc điểm đóng gói là độc hại (tương tự vấn đề 1).** |

* 1. **Công việc/tính năng/kỹ thuật mà nhóm thực hiện lập trình và triển khai cho demo:**

|  |
| --- |
| Nhóm thực hiện lại thí nghiệm 1 và thí nghiệm 4 đã đề cập ở **H**. Vì một số hạn chế về phần cứng nên nhóm không thể xử lý Lab dataset, chỉ triển khai được những thí nghiệm chỉ có sử dụng Wild dataset.  Sử dụng Wild dataset tác giả công bố thuộc định dạng .pickle và đọc thành dataframe. Tiền xử lý, chọn lọc tập train-test sao cho đúng tính chất của thí nghiệm gốc. Xây dựng mô hình với loạt các thuật toán có ứng dụng cây quyết định. Đánh giá bằng chi tiết nhiều chỉ số khác nhau.  Kết quả: *2 thí nghiệm đưa ra kết luận giống với bài báo.* |

* 1. **Các khó khăn, thách thức hiện tại khi thực hiện:**

|  |
| --- |
| Gặp vấn đề khi triển khai lại mã nguồn của tác giả công bố, nhưng đã tự thực hiện lại một số thí nghiệm tương tự bài báo dựa vào datasets của tác giả. |

1. **TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH SO VỚI KẾ HOẠCH THỰC HIỆN:**

|  |
| --- |
| **99%**: tuy không thể triển khai lại mã nguồn tác giả công bố, nhưng đã tự thực hiện lại một số thí nghiệm tương tự bài báo dựa vào datasets của tác giả. |

1. **NHẬT KÝ PHÂN CÔNG NHIỆM VỤ:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Phân công nhiệm vụ** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

BÁO CÁO TỔNG KẾT CHI TIẾT

**Phần bên dưới của báo cáo này là tài liệu báo cáo tổng kết - chi tiết của nhóm thực hiện cho đề tài này.**

*Qui định: Mô tả các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, số liệu thống kê trong bảng biểu, có giải thích)*

* 1. **Phương pháp thực hiện**

*<Trình bày kiến trúc, thành phần của hệ thống trong bài báo>*

*<Trình bày kiến trúc, thành phần đã thực hiện (nội dung mà nhóm đã thực hiện)>*

* 1. **Chi tiết cài đặt, hiện thực**

*<cách cài đặt, lập trình trên máy tính, cấu hình máy tính sử dụng, chuẩn bị dữ liệu, v.v>*

* 1. **Kết quả thực nghiệm**

*<mô tả hình ảnh về thực nghiệm, bảng biểu số liệu thống kê từ thực nghiệm, nhận xét về kết quả thu được.>*

* 1. **Hướng phát triển**

*<Nêu hướng phát triển tiềm năng của đề tài này trong tương lai. Nhận xét về tính ứng dụng của đề tài>.*

*Sinh viên báo cáo các nội dung mà nhóm đã thực hiện, có thể là 1 phần hoặc toàn bộ nội dung của bài báo. Nếu nội dung thực hiện có khác biệt với bài báo (như cấu hình, tập dữ liệu, kết quả,...), sinh viên cần chỉ rõ thêm khác biệt đó và nguyên nhân.*

---

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

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, 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.
* Đặt tên theo định dạng: [Mã lớp]-Project\_Final\_NhomX\_Madetai. (trong đó X và Madetai là mã số thứ tự nhóm và Mã đề tài trong danh sách đăng ký nhóm đồ án).

*Ví dụ: [NT521.N11.ANTT]-Project\_Final\_Nhom03\_CK01.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

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

**HẾT**