HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thu Trang

PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - NĂM 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Thu Trang

PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

TS. PHẠM HOÀNG DUY

HÀ NỘI - NĂM 2025

LỜI CAM ĐOAN

Tôi là Nguyễn Thu Trang, học viên cao học ngành Hệ thống thông tin. Tôi xin cam đoan đề án "Phát hiện mã độc nhúng trong file PDF" là do tôi nghiên cứu, tìm hiểu và phát triển dưới sự hướng dẫn của TS. Phạm Hoàng Duy, không phải sự sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày tháng năm 2026

Học viên

Nguyễn Thu Trang

LỜI CẢM ƠN

Để hoàn thành đề án này, bên cạnh sự nỗ lực của bản thân, em đã nhận được rất nhiều sự giúp đỡ và hỗ trợ quý báu từ các thầy cô và nhà trường.

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới thầy TS Phạm Hoàng Duy, người đã tận tình hướng dẫn, định hướng khoa học và luôn động viên, tạo điều kiện thuận lợi cho em trong suốt quá trình nghiên cứu và thực hiện đề án. Những ý kiến góp ý, chỉ dẫn chuyên môn và kinh nghiệm thực tiễn của thầy là nền tảng quan trọng giúp em hoàn thiện bài viết này.

Em cũng xin chân thành cảm ơn các thầy cô giáo công tác trong Học viện Công nghệ Bưu chính viễn thông, nhất là Quý thầy cô, cán bộ, giảng viên Khoa Công nghệ thông tin và Khoa Đào tạo Sau đại học đã trang bị cho em những kiến thức quý báu, đồng thời tạo điều kiện thuận lợi về cơ sở vật chất, tài liệu học tập và môi trường nghiên cứu chuyên nghiệp trong suốt thời gian học tập tại Học viện.

Mặc dù đã rất cố gắng trong quá trình nghiên cứu và hoàn thiện đề án, nhưng do hạn chế về thời gian, kinh nghiệm và kiến thức thực tiễn, bài viết khó tránh khỏi những thiếu sót. Em kính mong nhận được những ý kiến đóng góp của quý thầy cô để đề án được hoàn thiện hơn trong tương lai.

Em xin trân trọng cảm ơn!

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT v
DANH MỤC BẢNG vii
DANH MỤC HÌNH ẢNH viii
TÓM TẮT ĐỀ ÁN ix
I. MỞ ĐẦU 1
1. Lý do chọn đề tài 1
2. Tổng quan về vấn đề nghiên cứu 1
3. Mục đích nghiên cứu 2
4. Đối tượng và phạm vi nghiên cứu 2
5. Phương pháp nghiên cứu 3
II. NỘI DUNG 4
CHƯƠNG 1 - TỔNG QUAN PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE 4
1.1. Cơ sở lý thuyết về file PDF 4
1.2. Khái niệm và phân loại mã độc 6
1.3. Phương pháp phát hiện mã độc 12
1.4. Thách thức trong phát hiện mã độc 15
1.5. Kết luận chương 1 16
CHƯƠNG 2 - PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF 18
2.1. Giải pháp phát hiện mã độc nhúng trong file PDF 18
2.2. Trích xuất đặc trưng từ file PDF 20
2.3. Mô tả giải pháp 23
2.4. Kết luận chương 2 27
CHƯƠNG 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ 29
3.1. Thiết lập môi trường thực nghiệm 29
3.2. Quy trình huấn luyện 29
3.3. Đánh giá mô hình 33

3.4. Kết luận chương 3 41
III. KẾT LUẬN 42
1. Các kết quả đạt được của đề án 42
2. Nhận xét, đề xuất, khuyến nghị 42
3. Hướng phát triển 43
IV. DANH MỤC CÁC TÀI LIỆU THAM KHẢO 44

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AES	Advanced Encryption Standard	Chuẩn mã hóa nâng cao
API	Application Programming Interface	Giao diện lập trình ứng dụng
AUC	Area Under the Curve	Diện tích dưới đường cong ROC
CIC	Canadian Institute for Cybersecurity	Viện An ninh mạng Canada
CIC-Evasive-PDFMal2022	CIC Evasive PDF Malware 2022 Dataset	Tập dữ liệu mã độc PDF 2022 của CIC
CSV	Comma-Separated Values	Tập tin giá trị phân cách bằng dấu phẩy
CV	Cross-Validation	Kiểm chứng chéo
CVE	Common Vulnerabilities and Exposures	Danh sách lỗ hổng phổ biến
DLLs	Dynamic Link Libraries	Thư viện liên kết động
DNNs	Deep Neural Networks	Mạng nơ-ron sâu
DOCX	Microsoft Word Open XML Document	Tập tin văn bản Word
DT	Decision Tree	Cây quyết định
Ensemble	Ensemble Learning	Học tập hợp
Evasive	Evasive Techniques	Kỹ thuật né tránh phát hiện
Feature Importance	Feature Importance	Mức độ quan trọng của đặc trưng
FN	False Negative	Âm tính giả
FP	False Positive	Dương tính giả
GridSearchCV	Grid Search Cross-Validation	Tìm kiếm tham số theo lưới kết hợp kiểm chứng chéo

IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
IPS	Intrusion Prevention System	Hệ thống ngăn chặn xâm nhập
KNN	K-Nearest Neighbors	Thuật toán k láng giềng gần nhất
NaN	Not a Number	Giá trị không phải là số
obj	Object	Đối tượng
Obfuscation	Code Obfuscation	Kỹ thuật làm rối mã
OTP	One-Time Password	Mật khẩu dùng một lần
PAdES	PDF Advanced Electronic Signatures	Chữ ký điện tử nâng cao trong PDF
PCA	Principal Component Analysis	Phân tích thành phần chính
PDF	Portable Document Format	Định dạng tài liệu di động
PDFMal2022	Evasive PDF Malware 2022 Dataset	Tập dữ liệu mã độc PDF năm 2022
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RC4	Rivest Cipher 4	Thuật toán mã hóa dòng RC4
RF	Random Forest	Rừng ngẫu nhiên
ROC	Receiver Operating Characteristic	Đường cong đặc trưng hoạt động
Static Analysis	Static Analysis	Phân tích tĩnh
SVM	Support Vector Machine	Máy vector hỗ trợ
TN	True Negative	Âm tính đúng
TP	True Positive	Dương tính đúng
URL	Uniform Resource Locator	Định danh tài nguyên thống nhất
XLSX	Microsoft Excel Spreadsheet	Tập tin bảng tính Excel
zip	Compressed Archive File	Tập tin nén

DANH MỤC BẢNG

Bảng 2.1. Các đặc trưng tĩnh trích xuất từ file PDF 22
Bảng 2.2. So sánh đặc điểm các thuật toán học máy phổ biến 23
Bảng 3.1. Thống kê phân bố tập dữ liệu nghiên cứu (sau làm sạch) 30
Bảng 3.2. Các siêu tham số thử nghiệm trong Grid Search 32
Bảng 3.3. Kết quả thử nghiệm trên tập kiểm tra 34
Bảng 3.4. Kết quả thử nghiệm trên tập đánh giá.....34
Bảng 3.5. So sánh hiệu suất giữa các mô hình.....38
Bảng 3. 6. Kết quả phân tích nhạy của tham số 39

DANH MỤC HÌNH ẢNH

Hình 2.1. Quy trình phát hiện mã độc trong file PDF bằng học máy. 19
Hình 3. 1. Ma trận nhầm lẫn trên tập đánh giá.....36
Hình 3. 2. Đường cong ROC và AUC trêm tập đánh giá. 36
Hình 3.3. Top 20 đặc trưng quan trọng nhất theo mô hình Random Forest. 37
Hình 3.4. Biểu đồ so sánh F1-score theo tham số huấn luyện. 40

TÓM TẮT ĐỀ ÁN

Trong bối cảnh các cuộc tấn công mạng ngày càng tinh vi, việc phát hiện mã độc nhúng trong các tệp tài liệu, đặc biệt là định dạng PDF, đã trở thành một thách thức quan trọng đối với lĩnh vực an toàn thông tin. Đề án tập trung nghiên cứu và xây dựng mô hình học máy nhằm phát hiện mã độc nhúng trong file PDF thông qua việc phân tích các đặc trưng tĩnh của cấu trúc tập tin.

Đề án tiến hành khảo sát các phương pháp phát hiện mã độc hiện có, làm rõ các kỹ thuật khai thác phổ biến trong tệp PDF như JavaScript, file nhúng, các đối tượng ẩn, ... Sử dụng tập dữ liệu CIC-Evasive-PDFMal2022 để thực hiện trích xuất đặc trưng, gồm các thông tin về cấu trúc tệp (số lượng đối tượng, kích thước, javascript, action tự động…) và các thành phần nhúng. Các đặc trưng sau đó được xử lý, chuẩn hóa và đưa vào mô hình huấn luyện.

Mô hình được sử dụng trong đề tài là Random Forest, với việc tối ưu tham số bằng GridSearchCV để nâng cao hiệu quả phân loại. Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác cao, khả năng phân biệt tốt giữa các tệp PDF lành tính và độc hại. Đồng thời, đề án cũng so sánh với một số mô hình khác như Decision Tree, SVM để đánh giá mức độ hiệu quả.

Kết quả không chỉ góp phần khẳng định tính khả thi của việc áp dụng học máy trong phát hiện mã độc PDF, mà còn là nền tảng để phát triển các hệ thống giám sát và phòng chống tấn công qua tài liệu trong tương lai.

Từ khóa: Phát hiện mã độc, file PDF, học máy, Random Forest, đặc trưng tĩnh, trích xuất đặc trưng, CIC-Evasive-PDFMal2022..

I. MỞ ĐẦU

1. Lý do chọn đề tài

Với tính phổ biến và khả năng hỗ trợ nhiều thành phần như JavaScript, file đính kèm, liên kết ẩn…, định dạng PDF không chỉ được sử dụng rộng rãi trong trao đổi, lưu trữ thông tin mà còn trở thành mục tiêu hàng đầu của các cuộc tấn công mạng. Tin tặc ngày càng khai thác nhiều kỹ thuật tinh vi như mã hóa, làm rối mã (obfuscation), đa hình (polymorphism) để nhúng mã độc vào các file PDF, khiến việc phát hiện trở nên khó khăn hơn.

Theo báo cáo của Kaspersky năm 2023, số lượng các cuộc tấn công bằng file PDF độc hại đã tăng 23% so với năm trước, gây ra nhiều hậu quả nghiêm trọng như lộ lọt thông tin, phá hoại hệ thống hoặc chiếm quyền kiểm soát máy tính. Các giải pháp truyền thống như phân tích chữ ký hay hành vi tuy có hiệu quả với mã độc đã biết, nhưng thường thất bại trước các biến thể mới hoặc mã độc chưa được cập nhật trong cơ sở dữ liệu. Trong bối cảnh đó, việc nghiên cứu và áp dụng học máy vào phát hiện mã độc nhúng trong file PDF trở thành một hướng đi tiềm năng, nhờ khả năng tự học, phân loại dựa trên các đặc trưng phức tạp và phát hiện được cả những mối đe dọa chưa từng xuất hiện trước đó.

Đề tài "Phát hiện mã độc nhúng trong file PDF" không chỉ đáp ứng yêu cầu cấp thiết về đảm bảo an toàn thông tin, mà còn mang lại giá trị nghiên cứu trong lĩnh vực an ninh mạng, đặc biệt là phát hiện mối đe dọa thông minh trên nền tảng dữ liệu lớn và học máy hiện đại.

2. Tổng quan về vấn đề nghiên cứu

PDF là định dạng tài liệu được sử dụng rộng rãi và có khả năng tích hợp nhiều thành phần như văn bản, hình ảnh, mã script hoặc đối tượng nhúng. Chính sự linh hoạt này khiến PDF trở thành mục tiêu hấp dẫn cho các cuộc tấn công bằng mã độc. Những giải pháp truyền thống dựa trên chữ ký hoặc quy tắc phát hiện chỉ phù hợp với các mẫu đã quen thuộc, nhưng lại thiếu hiệu quả trước các biến thể mã độc được làm mới hoặc che giấu bằng kỹ thuật làm rối. Điều này cho thấy sự cần thiết của những phương pháp phát hiện tiên tiến hơn.

Nhiều nghiên cứu trước đã áp dụng phân tích tĩnh và phân tích động vào việc phát hiện mã độc trong file PDF. Tuy vậy, các phương pháp học máy hiện nay vẫn còn đối mặt với không ít thách thức, chẳng hạn như khó xử lý các tệp PDF có cấu trúc phức tạp hoặc gặp khó khăn khi phân biệt những mẫu mã độc được che giấu một cách tinh vi (evasive techniques) nhằm tránh bị hệ thống phát hiện.

3. Mục đích nghiên cứu

Mục đích chính của đề án tốt nghiệp là xây dựng một mô hình học máy hiệu quả để phát hiện mã độc nhúng trong file PDF. Đề tài tập trung vào việc sử dụng các kỹ thuật học máy, đặc biệt là phân tích các đặc trưng tĩnh (Static Analysis) của file PDF, nhằm nâng cao độ chính xác và khả năng phát hiện các mã độc mới hoặc chưa được biết đến.

Về mặt lý luận, nghiên cứu sẽ góp phần mở rộng kiến thức về ứng dụng học máy trong lĩnh vực an ninh mạng, đặc biệt trong việc phát hiện mã độc. Đề tài cũng sẽ phát triển cơ sở lý thuyết về các đặc trưng của file PDF có thể liên quan đến mã độc và cách sử dụng học máy để phân loại và phát hiện mã độc dựa trên các đặc trưng này.

Về mặt thực tiễn, kết quả của đề án này sẽ giúp cải thiện hiệu quả phát hiện mã độc trong các hệ thống trao đổi tài liệu sử dụng file PDF, từ đó nâng cao an toàn thông tin cho người dùng. Hệ thống phát hiện mã độc này có thể ứng dụng rộng rãi trong các tổ chức, công ty và nền tảng giao tiếp trực tuyến, giảm thiểu nguy cơ bị tấn công qua các file PDF độc hại.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là các file PDF chứa mã độc nhúng, cụ thể là các file PDF có thể bị tấn công thông qua việc nhúng các mã độc như virus, trojan, hoặc các đoạn mã JavaScript có hại. Các file PDF này được sử dụng trong môi trường trao đổi tài liệu trực tuyến và có thể được tải xuống từ các nguồn không xác định, gây nguy cơ lớn cho người dùng và hệ thống.

Phạm vi nghiên cứu tập trung vào việc xây dựng mô hình học máy có khả năng phân loại file PDF theo mức độ an toàn, dựa trên đặc trưng tĩnh của cấu trúc tập tin. Trong khuôn khổ đề tài, bộ dữ liệu CIC-Evasive-PDFMal2022 được sử dụng làm nguồn dữ liệu thực nghiệm để huấn luyện và đánh giá mô hình. Nghiên cứu chưa mở rộng sang các định dạng văn bản khác như DOCX hoặc XLSX.

5. Phương pháp nghiên cứu

Thu thập dữ liệu và phân tích đặc trưng: Xử lý dữ liệu từ bộ CIC-Evasive-PDFMal2022, tiến hành tiền xử lý nhằm loại bỏ nhiễu và chuẩn hóa dữ liệu, sau đó lựa chọn các đặc trưng liên quan như kích thước file (PdfSize), mã JavaScript, các hành động tự động (OpenAction, AA) hoặc các đối tượng nhúng. Những đặc trưng này sẽ được sử dụng để huấn luyện mô hình học máy và đánh giá độ chính xác thông qua các chỉ số đo lường.

Phát triển mô hình học máy: Lựa chọn thuật toán Random Forest, chia dữ liệu thành tập huấn luyện, kiểm tra và đánh giá, huấn luyện mô hình bằng tập huấn luyện, điều chỉnh các tham số (n_estimators, max_depth) bằng kỹ thuật Grid Search Cross-Validation.

Thử nghiệm và đánh giá: Sử dụng các chỉ số đánh giá phổ biến như accuracy, precision, recall, F1-score, để so sánh hiệu quả của mô hình với các phương pháp hiện có.

II. NỘI DUNG

CHƯƠNG 1 - TỔNG QUAN PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE

Giới thiệu: *Chương này cung cấp cái nhìn tổng quan về vấn đề mã độc nhúng trong file PDF, bao gồm các khái niệm cơ bản về định dạng PDF, các loại mã độc thường gặp và các kỹ thuật khai thác liên quan. Đồng thời, chương nêu lên những hạn chế của phương pháp phát hiện mã độc truyền thống và các thách thức cần giải quyết.*

1.1. Cơ sở lý thuyết về file PDF

Định dạng tài liệu di động (PDF - Portable Document Format) là một định dạng tài liệu được phát triển bởi Adobe vào năm 1993, nhằm mục đích duy trì định dạng và nội dung của tài liệu bất kể phần mềm, phần cứng hay hệ điều hành được sử dụng. PDF cho phép tích hợp văn bản, hình ảnh, đồ họa, liên kết và thậm chí là các tập lệnh (script) vào cùng một file duy nhất.

Một file PDF bao gồm bốn thành phần chính:

- Phần đầu tệp (Header**)** chứa thông tin nhận diện phiên bản của PDF. Dòng đầu tiên của file PDF luôn có định dạng %PDF-x.y, trong đó x.y là phiên bản của file PDF, ví dụ %PDF-1.7 cho phiên bản 1.7. Hiện nay, các phiên bản PDF mới nhất (ví dụ: PDF 2.0) đã bổ sung thêm nhiều tính năng bảo mật như hỗ trợ chữ ký số nâng cao (PAdES) và mã hóa mạnh hơn.
- Thân tệp (Body) chứa nội dung chính của tài liệu, bao gồm văn bản, hình ảnh, font chữ, đối tượng đa phương tiện như video, âm thanh hoặc hình ảnh động, cùng các yếu tố tương tác như biểu mẫu hoặc script có thể thực thi. Các đối tượng này có thể được lưu trữ trực tiếp trong Body hoặc được nén và lưu trữ trong các Object Stream. Content Stream chứa các lệnh vẽ (drawing commands) để hiển thị nội dung của trang PDF.
- Bảng tham chiếu chéo (Cross-reference Table) giúp trình đọc PDF nhanh chóng truy xuất đến các đối tượng trong file mà không cần quét toàn bộ nội

dung. Mỗi dòng trong bảng này đại diện cho một đối tượng và cung cấp thông tin về vị trí của nó trong file.

- Phần kết (Trailer) là phần kết thúc của file chứa thông tin cần thiết để đọc file PDF, bao gồm đối tượng gốc (Root Object) – điểm bắt đầu của cấu trúc tài liệu, số lượng đối tượng trong file, byte offset của bảng Xref và ID của tài liệu để kiểm tra tính toàn

Các stream dữ liệu trong file PDF thường được nén bằng các thuật toán như FlateDecode hoặc LZWDecode để giảm kích thước file. Ngoài ra, file PDF cũng có thể được mã hóa bằng các thuật toán như RC4 hoặc AES để bảo vệ nội dung khỏi việc truy cập trái phép. Việc giải nén và giải mã các stream này là cần thiết để phân tích nội dung bên trong, nhưng đòi hỏi các công cụ và kỹ thuật chuyên biệt.

Mặc dù PDF là một định dạng tài liệu an toàn, nhưng nó có thể chứa các thành phần nguy hiểm có thể bị khai thác để phát tán mã độc. Một số thành phần nguy hiểm phổ biến bao gồm:

- JavaScript là một trong những yếu tố nguy hiểm phổ biến. PDF hỗ trợ JavaScript để tăng cường tính tương tác, nhưng kẻ tấn công có thể nhúng mã độc vào file nhằm tự động thực thi mã khi mở file, thu thập dữ liệu từ người dùng hoặc khai thác lỗ hổng trình đọc PDF để thực thi mã độc. Các API JavaScript nguy hiểm thường được sử dụng trong mã độc PDF bao gồm mở một URL bên ngoài (app.launchURL), in dữ liệu ra màn hình (util.printd) và gửi dữ liệu biểu mẫu (this.submitForm).
- Embedded Files cũng là một rủi ro đáng kể. PDF có thể nhúng các file khác như .exe, .zip, .docx dưới dạng tệp đính kèm. Những file này có thể chứa phần mềm độc hại và dễ bị lợi dụng trong các cuộc tấn công phishing.
- Ngoài ra, PDF hỗ trợ Action (hành động tự động) thông qua lệnh OpenAction, cho phép thực hiện các hành vi như chạy JavaScript ngay khi mở file, chuyển hướng người dùng đến một trang web độc hại hoặc tự động tải xuống và chạy file từ internet.

1.2. Khái niệm và phân loại mã độc

1.2.1. Khái niệm mã độc

Mã độc (malware) xuất hiện từ rất sớm trong lịch sử phát triển của máy tính. Một trong những chương trình đầu tiên có khả năng tự di chuyển giữa các hệ thống là Creeper, được thử nghiệm vào đầu thập niên 1970 trên mạng ARPANET. Creeper có thể sao chép sang các máy khác nhau và hiển thị thông báo mang tính thách thức: "*I'm the Creeper, catch me if you can*." Dù không mang tính phá hoại, Creeper được xem là tiền thân của các phần mềm tự lan truyền trong những thập kỷ tiếp theo. Sau đó, vào đầu những năm 1980, Elk Cloner xuất hiện trên máy tính Apple II và là một trong những virus lan rộng đầu tiên được ghi nhận, đánh dấu giai đoạn phần mềm độc hại bắt đầu trở thành mối đe dọa thực sự đối với người dùng.

Mã độc nói chung được hiểu là những chương trình hoặc đoạn mã được tạo ra với mục đích xâm nhập trái phép, phá hoại tài nguyên, đánh cắp dữ liệu hoặc làm gián đoạn hoạt động hệ thống. Tùy theo cách thức lan truyền và phương pháp gây hại, mã độc có thể được phân loại thành nhiều nhóm như virus, worm, trojan, rootkit, botnet hay ransomware. Dù thuộc loại nào, mã độc đều tác động đến ba yếu tố cốt lõi của an toàn thông tin: tính bí mật, tính toàn vẹn, và tính sẵn sàng.

Mã độc trong file PDF là các đoạn mã hoặc nội dung độc hại được nhúng vào bên trong tệp PDF nhằm khai thác lỗ hổng của các trình đọc PDF hoặc lừa người dùng thực thi hành vi trái phép. Do PDF hỗ trợ nhiều tính năng mạnh mẽ như nhúng JavaScript, chèn file đính kèm và tự động thực thi lệnh, tin tặc có thể lợi dụng điều này để tấn công hệ thống.

Mã độc PDF nguy hiểm bởi nó thường được ngụy trang tinh vi, khiến người dùng tin rằng họ đang mở một tài liệu vô hại. Vì thế, việc phát hiện và phân tích các mã độc ẩn trong PDF đóng vai trò quan trọng trong bảo mật hệ thống thông tin.

1.2.2. Các loại mã độc thường gặp trong file PDF

File PDF không chỉ đơn thuần là một định dạng tài liệu tĩnh mà còn hỗ trợ nhiều tính năng mạnh mẽ như nhúng JavaScript, liên kết tệp tin và mã hóa dữ liệu. Tuy nhiên, chính những đặc điểm này đã trở thành công cụ để tin tặc lợi dụng,

nhúng mã độc nhằm thực hiện các cuộc tấn công tinh vi. Việc nhận diện và phân tích các loại mã độc này đóng vai trò quan trọng trong công tác bảo mật thông tin và phòng chống tấn công mạng. Một số loại mã độc điển hình trong file PDF:

Mã độc JavaScript

Mã độc JavaScript trong file PDF là một trong những kỹ thuật tấn công phổ biến mà tin tặc sử dụng để khai thác người dùng. JavaScript thường được tích hợp trong PDF để tạo các chức năng động, như tự động điền biểu mẫu, hiển thị nội dung tương tác hoặc kiểm tra tính hợp lệ của dữ liệu nhập vào. Tuy nhiên, chính khả năng này cũng trở thành điểm yếu để hacker lợi dụng, nhúng mã độc vào file PDF và khai thác hệ thống của người dùng khi họ mở tài liệu mà không hề hay biết.

Khi một file PDF có chứa mã JavaScript độc hại được mở, mã này có thể thực hiện hàng loạt hành vi nguy hiểm như đánh cắp dữ liệu cá nhân, tự động tải xuống và chạy phần mềm độc hại, thu thập thông tin đăng nhập hoặc thậm chí chiếm quyền điều khiển máy tính của nạn nhân. Một trong những cách thức phổ biến là mã độc JavaScript được sử dụng để hiển thị các cửa sổ giả mạo yêu cầu người dùng nhập mật khẩu hoặc thông tin tài chính. Nếu nạn nhân mắc bẫy và cung cấp thông tin, dữ liệu này sẽ ngay lập tức được gửi về máy chủ của hacker.

Ngoài ra, một số cuộc tấn công lợi dụng JavaScript để kích hoạt lỗ hổng bảo mật trong phần mềm đọc PDF, khiến hệ thống của người dùng bị nhiễm virus hoặc Trojan mà họ không hề nhận ra. Lỗ hổng CVE-2009-4324 là một trong những ví dụ điển hình về cách mã JavaScript độc hại trong file PDF có thể khai thác phần mềm Adobe Reader để thực thi mã từ xa.

Mã khai thác lỗ hổng bảo mật (Exploit)

Mã khai thác lỗ hổng bảo mật trong file PDF là một trong những phương thức tấn công nguy hiểm nhất, vì nó có thể hoạt động ngay cả khi người dùng không thực hiện bất kỳ thao tác nào khác ngoài việc mở file. Exploit lợi dụng các lỗ hổng trong phần mềm đọc PDF, chẳng hạn như Adobe Reader, Foxit Reader, hoặc các trình đọc PDF mặc định trên hệ điều hành, để thực thi mã độc trên máy tính nạn nhân mà không cần sự cho phép.

Thông thường, hacker sẽ tạo ra một file PDF trông có vẻ vô hại nhưng thực chất đã được nhúng mã khai thác. Khi nạn nhân mở file này, mã khai thác sẽ kích hoạt lỗ hổng trong phần mềm đọc PDF và cho phép hacker thực hiện nhiều hành vi nguy hiểm như cài đặt phần mềm gián điệp, chiếm quyền điều khiển hệ thống từ xa, hoặc đánh cắp dữ liệu cá nhân.

Một số lỗ hổng từng bị hacker khai thác rộng rãi bao gồm CVE-2013-2729 và CVE-2018-4990, trong đó hacker đã lợi dụng các điểm yếu trong Adobe Reader để thực thi mã từ xa và cài đặt phần mềm độc hại trên máy tính của người dùng. Các cuộc tấn công kiểu này rất nguy hiểm vì ngay cả những người dùng cẩn thận cũng có thể bị lừa mở một file PDF tưởng chừng như hợp lệ nhưng thực chất lại chứa mã độc.

Trojan

Trojan PDF là một loại mã độc nguy hiểm được ngụy trang dưới dạng một file PDF hợp lệ, khiến người dùng không nghi ngờ mà mở nó. Khi file này được kích hoạt, mã độc bên trong sẽ âm thầm thực hiện các hoạt động trái phép như tạo cửa hậu (backdoor) để hacker có thể truy cập từ xa vào hệ thống của nạn nhân, theo dõi hoạt động bàn phím để đánh cắp mật khẩu, hoặc thậm chí biến máy tính của người dùng thành một phần của mạng botnet phục vụ các cuộc tấn công mạng lớn hơn.

Một ví dụ điển hình của loại mã độc này là Trojan.Pidief, một trojan khai thác lỗ hổng trong Adobe Reader để chạy mã độc mà không cần sự tương tác của người dùng. Khi file PDF chứa Trojan này được mở, nó có thể ngay lập tức tải xuống thêm phần mềm độc hại khác, làm lây nhiễm toàn bộ hệ thống mà người dùng không hề hay biết. Trojan PDF có thể sử dụng các kỹ thuật rootkit để che giấu sự hiện diện của chúng trên hệ thống, khiến chúng khó bị phát hiện bởi các phần mềm diệt virus.

Phishing

Tấn công lừa đảo thông qua tệp PDF là kỹ thuật phổ biến mà kẻ tấn công sử dụng để chiếm đoạt thông tin nhạy cảm của người dùng, đặc biệt là dữ liệu đăng

nhập ngân hàng, email hoặc tài khoản mạng xã hội. Trong kiểu tấn công này, tin tặc tạo ra các tệp PDF trông hoàn toàn bình thường nhưng được nhúng các liên kết dẫn đến những trang web giả mạo, có giao diện sao chép gần giống trang chính thức nhằm đánh lừa người xem.

Khi mở tệp và nhấp vào các liên kết được chèn bên trong, nạn nhân sẽ được chuyển đến các trang web lừa đảo, nơi họ có thể bị yêu cầu cung cấp thông tin đăng nhập, số thẻ ngân hàng, mã bảo mật hoặc OTP. Một số dạng PDF phishing còn thiết kế các biểu mẫu trực tiếp trong tài liệu, khiến người dùng tưởng rằng họ đang thực hiện thao tác hợp lệ và tự nguyện cung cấp dữ liệu cá nhân.

Các tệp PDF dạng phishing thường xuất hiện trong email giả mạo có nội dung khẩn cấp, chẳng hạn như yêu cầu xác minh tài khoản từ ngân hàng, cảnh báo bảo mật từ dịch vụ thanh toán, hoặc thông báo từ các tổ chức uy tín. Điều này khiến người dùng dễ mất cảnh giác và trở thành mục tiêu bị đánh cắp thông tin..

Tấn công thông qua file nhúng (Embedded Files Attack)

Một trong những phương thức tấn công tinh vi khác là nhúng các file độc hại vào bên trong một file PDF và lừa người dùng tải xuống hoặc mở các tệp này. Những file nhúng có thể là tệp thực thi (.exe, .bat), tài liệu Office có macro độc hại (.docm, .xlsm), hoặc file nén (.zip, .rar) chứa virus. Khi người dùng mở các file này, mã độc sẽ ngay lập tức được kích hoạt, gây lây nhiễm vào hệ thống hoặc cho phép hacker truy cập vào dữ liệu cá nhân của nạn nhân. Phương thức tấn công này thường được sử dụng trong các cuộc tấn công có chủ đích nhằm vào doanh nghiệp hoặc cá nhân có dữ liệu quan trọng.

1.2.3. Kỹ thuật nhúng mã độc vào file PDF

Mã độc trong file PDF là một mối đe dọa nghiêm trọng vì nó có thể dễ dàng lây nhiễm vào hệ thống mà người dùng không hề hay biết. Hacker sử dụng nhiều kỹ thuật để nhúng mã độc vào file PDF nhằm khai thác lỗ hổng bảo mật, lừa đảo người dùng hoặc thực thi các hành vi xâm nhập hệ thống. Các kỹ thuật phổ biến được tin tặc sử dụng để nhúng mã độc vào file PDF.

Nhúng mã JavaScript độc hại

Một kỹ thuật thường được kẻ tấn công sử dụng để cài mã độc vào tệp PDF là lợi dụng khả năng hỗ trợ JavaScript của các trình đọc PDF. Nhiều phần mềm như Adobe Acrobat cho phép chạy JavaScript nhằm tự động hóa các tác vụ như xử lý biểu mẫu, kiểm tra dữ liệu hoặc tương tác với người dùng. Chính cơ chế hợp pháp này lại trở thành điểm yếu mà tin tặc khai thác bằng cách chèn các đoạn script độc hại vào trong tài liệu.

Mã JavaScript được nhúng có thể hoạt động theo nhiều cách khác nhau tùy thuộc vào mục tiêu tấn công. Chúng có thể thu thập thông tin về hệ thống, mở kết nối ra bên ngoài để tải xuống các tệp độc hại khác hoặc kích hoạt lỗ hổng trong trình đọc PDF để chiếm quyền điều khiển. Một số script được thiết kế để âm thầm cài đặt trojan, keylogger hoặc thậm chí ransomware mà người dùng không hề hay biết.

Ngoài ra, kẻ tấn công còn có thể sử dụng JavaScript trong PDF để tạo các hộp thoại hoặc giao diện giả mạo, yêu cầu người dùng nhập thông tin nhạy cảm như mật khẩu hoặc dữ liệu cá nhân. Những thông tin này sau đó được gửi trực tiếp về máy chủ của tin tặc. Nhờ khả năng che giấu khéo léo trong tệp PDF, các đoạn mã JavaScript độc hại trở thành mối đe dọa nguy hiểm đối với người dùng thiếu cảnh giác..

Khai thác lỗ hổng bảo mật trong phần mềm đọc PDF (Exploit)

Thay vì chèn mã độc trực tiếp vào nội dung file PDF, hacker có thể thiết kế một tập tin PDF đặc biệt để khai thác lỗ hổng bảo mật trong phần mềm đọc PDF. Khi nạn nhân mở tài liệu, phần mềm sẽ gặp lỗi và thực thi mã độc mà hacker đã nhúng sẵn.

Những lỗ hổng bảo mật thường bị khai thác là lỗi tràn bộ đệm (Buffer Overflow), lỗi thực thi mã từ xa (Remote Code Execution), lỗi xử lý hình ảnh hoặc font chữ trong PDF. Một số cuộc tấn công trước đây đã lợi dụng lỗ hổng trong Adobe Reader, cho phép hacker kiểm soát hoàn toàn hệ thống của nạn nhân chỉ bằng cách mở một file PDF độc hại.

Các cuộc tấn công dạng này thường không yêu cầu người dùng thực hiện thêm thao tác nào, chỉ cần mở file PDF là mã độc có thể được kích hoạt ngay lập tức. Do đó, đây là một trong những phương pháp nguy hiểm nhất vì nó có thể tấn công ngay cả những người dùng cẩn trọng.

Chèn mã độc vào file đính kèm bên trong PDF

File PDF không chỉ chứa văn bản và hình ảnh mà còn có thể đính kèm các tập tin khác như Word, Excel, PowerPoint, hoặc tệp thực thi (.exe, .bat, .vbs). Hacker có thể nhúng một file chứa mã độc vào PDF và sử dụng các thủ thuật xã hội để dụ dỗ nạn nhân mở file đính kèm.

Khi người dùng mở tập tin đính kèm, các macro độc hại (trong file Word, Excel) hoặc phần mềm độc hại sẽ được kích hoạt. Một số phương thức tấn công phổ biến bao gồm:

- Sử dụng macro trong file Word/Excel để tải xuống và cài đặt mã độc từ máy chủ của hacker.
- Nhúng file .exe hoặc script thực thi vào PDF để tự động khởi chạy chương trình độc hại khi nạn nhân nhấp vào.
- Đánh lừa người dùng bằng cách đổi đuôi file (ví dụ: "document.pdf.exe" nhưng hiển thị dưới dạng "document.pdf" để người dùng nhầm lẫn).

Nhúng mã độc vào hình ảnh hoặc font chữ trong PDF

Kỹ thuật này tinh vi hơn và thường khó bị phát hiện bởi các phần mềm diệt virus. Hacker có thể sử dụng Steganography (giấu mã độc trong dữ liệu khác) để nhúng mã độc vào hình ảnh hoặc font chữ trong PDF. Khi phần mềm đọc PDF xử lý hình ảnh hoặc font chữ này, mã độc sẽ được giải mã và thực thi trên hệ thống.

Một số cuộc tấn công trong quá khứ đã khai thác lỗ hổng trong font chữ TrueType (TTF) hoặc OpenType (OTF) trong PDF. Khi phần mềm đọc PDF tải font chữ này để hiển thị nội dung, nó sẽ vô tình kích hoạt mã độc mà hacker đã chèn vào. Kỹ thuật này giúp mã độc qua mặt được các phần mềm quét virus truyền thống, vì nó không chứa mã độc rõ ràng trong nội dung file mà được giấu trong các phần tử không ngờ tới như hình ảnh hoặc font chữ.

PDF lừa đảo (Phishing PDF)

Một dạng tấn công phổ biến khác là sử dụng file PDF để thực hiện các cuộc tấn công lừa đảo (phishing). Hacker sẽ tạo ra một file PDF có chứa liên kết giả mạo hoặc biểu mẫu thu thập thông tin cá nhân, sau đó gửi file này qua email hoặc các nền tảng nhắn tin.

Tài liệu PDF có thể được thiết kế để trông giống như một hóa đơn, thông báo từ ngân hàng, hoặc tài liệu công việc quan trọng, trong đó chứa một liên kết dẫn đến một trang web giả mạo. Khi người dùng nhấp vào liên kết và nhập thông tin đăng nhập, dữ liệu sẽ bị hacker thu thập và sử dụng cho các mục đích xấu.

Một số file PDF phishing còn sử dụng nút bấm hoặc biểu mẫu tương tác ngay trong tài liệu để thu thập thông tin trực tiếp từ nạn nhân mà không cần chuyển hướng đến trang web bên ngoài. Điều này làm cho các cuộc tấn công trở nên tinh vi và khó phát hiện hơn.

1.3. Phương pháp phát hiện mã độc

Trước sự gia tăng nhanh chóng của các cuộc tấn công thông qua file PDF, nhiều phương pháp phát hiện mã độc đã được đề xuất và phát triển nhằm nâng cao khả năng bảo vệ hệ thống thông tin. Các phương pháp này khác nhau về mức độ phân tích, chi phí triển khai và hiệu quả phát hiện, nhưng nhìn chung có thể phân thành ba nhóm chính gồm phân tích tĩnh, phân tích động và các phương pháp dựa trên học máy. Việc khảo sát các hướng tiếp cận này cho thấy không tồn tại một phương pháp đơn lẻ nào có thể giải quyết triệt để bài toán phát hiện mã độc PDF, mà mỗi phương pháp chỉ phù hợp với những bối cảnh và yêu cầu cụ thể.

1.3.1. Phân tích tĩnh (Static Analysis)

Phân tích tĩnh là phương pháp tiếp cận và phân tích file PDF bằng cách kiểm tra cấu trúc, mã nguồn (như JavaScript, Action) và các siêu dữ liệu (metadata) mà không cần thực thi file. Nguyên lý hoạt động dựa trên việc quét các thành phần bên trong file để tìm kiếm các dấu hiệu bất thường (anomalies) hoặc các đoạn mã nguy hiểm đã biết.

Các công cụ phân tích tĩnh thường thực hiện việc bóc tách (parsing) cấu trúc PDF để trích xuất các từ khóa (keywords) nhạy cảm như /JavaScript, /OpenAction, /Launch, hoặc các đối tượng nhúng đáng ngờ. Ưu điểm lớn nhất của phương pháp này là tốc độ xử lý nhanh, an toàn (do không chạy mã độc) và có khả năng bao quát toàn bộ mã nguồn của file. Đối với bài toán phát hiện mã độc quy mô lớn, phân tích tĩnh thường là bước lọc đầu tiên và quan trọng nhất.

Tuy nhiên, phân tích tĩnh truyền thống (dựa trên chữ ký) có nhược điểm là khó phát hiện các mã độc sử dụng kỹ thuật làm rối (obfuscation) hoặc mã hóa payload để che giấu nội dung thực sự. Để khắc phục điều này, các phương pháp hiện đại (như trong đề tài này) kết hợp phân tích tĩnh với Học máy (Machine Learning) để nhận diện các mẫu cấu trúc bất thường thay vì chỉ tìm kiếm từ khóa đơn thuần.

1.3.2. Phân tích động (Dynamic Analysis)

Phân tích động tiếp cận bài toán bằng cách thực thi file PDF trong môi trường cách ly và quan sát hành vi phát sinh trong quá trình chạy. Phương pháp này cho phép ghi nhận các hành vi như thực thi JavaScript, truy cập bộ nhớ, gọi API hệ thống hoặc thiết lập kết nối mạng, từ đó đánh giá mức độ nguy hiểm của file.

Nhiều công trình nghiên cứu đã sử dụng các hệ thống sandbox để mở file PDF và thu thập đặc trưng hành vi phục vụ cho việc phát hiện mã độc. Các kết quả thực nghiệm cho thấy phân tích động có khả năng phát hiện hiệu quả các exploit khai thác lỗ hổng trong phần mềm đọc PDF, ngay cả khi mã độc được che giấu trong cấu trúc file.

Mặc dù mang lại khả năng phát hiện tốt hơn so với phân tích tĩnh, phân tích động cũng tồn tại nhiều hạn chế. Quá trình thực thi và giám sát hành vi đòi hỏi chi phí tính toán lớn và thời gian phân tích dài. Bên cạnh đó, nhiều mã độc hiện đại được thiết kế để trì hoãn hành vi hoặc né tránh môi trường sandbox, làm giảm độ tin cậy của kết quả phân tích. Do đó, phương pháp này thường được sử dụng như một bước phân tích chuyên sâu thay vì giải pháp độc lập.

1.3.3. Sử dụng học máy (Machine Learning-based Detection)

Trước những hạn chế của phân tích tĩnh và phân tích động, các phương pháp dựa trên học máy đã trở thành hướng nghiên cứu chủ đạo trong bài toán phát hiện mã độc PDF. Thay vì phụ thuộc vào chữ ký cố định hoặc quan sát hành vi đơn lẻ, học máy cho phép mô hình tự động học các đặc trưng phân biệt giữa file PDF lành tính và file độc hại từ dữ liệu huấn luyện.

Nhiều nghiên cứu đã sử dụng các đặc trưng trích xuất từ cấu trúc file PDF như số lượng object, sự hiện diện của JavaScript, action tự động và file nhúng để huấn luyện các mô hình phân loại. Các thuật toán như Naive Bayes, Support Vector Machine, Decision Tree và Random Forest được áp dụng rộng rãi và cho thấy hiệu quả vượt trội so với các phương pháp dựa trên chữ ký truyền thống. Trong đó, Random Forest thường đạt hiệu năng ổn định nhờ khả năng kết hợp nhiều cây quyết định, giảm hiện tượng quá khớp (overfitting và xử lý tốt dữ liệu có chiều cao và phân bố không đồng đều.

Bên cạnh các phương pháp dựa trên đặc trưng tĩnh, một số công trình còn kết hợp học máy với phân tích động hoặc sử dụng các mô hình học sâu để khai thác đặc trưng hành vi. Các hướng tiếp cận này cho thấy tiềm năng lớn trong việc phát hiện các mẫu mã độc mới và chưa từng xuất hiện trong dữ liệu huấn luyện. Tuy nhiên, hiệu quả của mô hình học máy phụ thuộc mạnh vào chất lượng dữ liệu, sự cân bằng giữa các lớp và cách lựa chọn đặc trưng. Ngoài ra, yêu cầu về khả năng giải thích và chi phí triển khai cũng là những yếu tố cần được cân nhắc khi áp dụng mô hình vào các hệ thống phát hiện mã độc PDF.

Từ khảo sát trên có thể thấy rằng các phương pháp dựa trên học máy, đặc biệt là các mô hình ensemble như Random Forest, mang lại sự cân bằng hợp lý giữa độ chính xác, tính ổn định và khả năng triển khai. Đây là hướng tiếp cận phù hợp để xây dựng hệ thống phát hiện mã độc PDF có hiệu quả, đồng thời tạo cơ sở khoa học cho việc lựa chọn mô hình và các tiêu chí đánh giá được trình bày trong các chương tiếp theo.

1.4. Thách thức trong phát hiện mã độc

Mã độc nhúng trong file PDF là mối đe dọa phổ biến trong nhiều môi trường ứng dụng khác nhau như hệ thống trao đổi tài liệu nội bộ, nền tảng chia sẻ file trực tuyến, cổng dịch vụ công, cũng như các hệ thống phân phối tài liệu qua web. Sự linh hoạt của định dạng PDF, kết hợp với mức độ tin cậy cao của người dùng đối với loại tài liệu này, khiến PDF trở thành phương tiện thuận lợi để che giấu và phát tán mã độc.

1.4.1. Đa dạng phương thức tấn công

File PDF hỗ trợ nhiều cơ chế tương tác và nội dung động, cho phép mã độc được triển khai dưới nhiều hình thức khác nhau như chèn JavaScript, lợi dụng các hành động tự động khi mở tài liệu, hoặc nhúng các tệp và liên kết tải mã độc từ xa. Các file PDF độc hại thường được thiết kế để hiển thị nội dung hợp lệ, khiến người dùng khó nhận biết dấu hiệu bất thường.

Ví dụ, trong các hệ thống chia sẻ tài liệu trực tuyến hoặc kho lưu trữ nội bộ, file PDF có thể chứa các đoạn JavaScript được kích hoạt khi người dùng mở hoặc cuộn trang, từ đó tải thêm mã độc hoặc khai thác lỗ hổng của phần mềm đọc PDF. Những file này thường không chứa tệp thực thi trực tiếp, làm cho việc phát hiện dựa trên cấu trúc trở nên khó khăn.

1.4.2. Kỹ thuật che giấu và vượt qua kiểm tra bảo mật

Các kỹ thuật che giấu là yếu tố then chốt giúp mã độc PDF vượt qua các cơ chế kiểm tra bảo mật. Nội dung độc hại thường được mã hóa, nén hoặc phân tách thành nhiều phần để tránh bị phát hiện bởi các công cụ phân tích tĩnh.

Trong nhiều trường hợp, mã JavaScript chỉ được giải mã tại thời điểm thực thi và có thể phụ thuộc vào môi trường hoặc phần mềm đọc PDF cụ thể. Điều này khiến các công cụ phân tích động gặp khó khăn nếu thời gian giám sát không đủ dài hoặc môi trường phân tích không phù hợp. Các kỹ thuật như trì hoãn thực thi hoặc phát hiện môi trường giả lập ngày càng được sử dụng để làm giảm hiệu quả của phân tích hành vi.

1.4.3. Hạn chế của các phương pháp phát hiện truyền thống

Các phương pháp phát hiện truyền thống, bao gồm phát hiện dựa trên chữ ký và luật heuristic, thường chỉ phù hợp với các mẫu mã độc đã được biết trước. Khi cấu trúc file hoặc mã độc được biến đổi, các phương pháp này dễ bị qua mặt.

Trong các hệ thống xử lý số lượng lớn tài liệu PDF, chẳng hạn như cổng tải tài liệu, nền tảng lưu trữ đám mây hoặc hệ thống quản lý tài liệu doanh nghiệp, yêu cầu về tốc độ xử lý khiến việc áp dụng phân tích động toàn diện trở nên khó khả thi. Điều này dẫn đến việc nhiều hệ thống chỉ có thể áp dụng các phương pháp kiểm tra nhanh, chấp nhận đánh đổi khả năng phát hiện các mẫu mã độc phức tạp..

1.4.4. Thách thức trong ứng dụng học máy để phát hiện mã độc PDF

Học máy được xem là hướng tiếp cận tiềm năng nhằm nâng cao hiệu quả phát hiện mã độc PDF, tuy nhiên việc triển khai các mô hình học máy trong thực tế vẫn đối mặt với nhiều thách thức. Dữ liệu huấn luyện thường có sự mất cân bằng đáng kể giữa file PDF lành tính và file độc hại, đặc biệt trong các hệ thống nơi PDF chủ yếu được sử dụng cho mục đích hợp pháp.

Bên cạnh đó, mô hình cần đảm bảo khả năng tổng quát tốt để phát hiện các mẫu mã độc mới, đồng thời duy trì hiệu năng xử lý phù hợp với các hệ thống yêu cầu phản hồi nhanh. Việc lựa chọn thuật toán, đặc trưng đầu vào và các tiêu chí đánh giá phù hợp đóng vai trò quan trọng trong việc xây dựng hệ thống phát hiện có tính ứng dụng cao.

1.5. Kết luận chương 1

Chương 1 đã trình bày tổng quan về mã độc nhúng trong file PDF, bao gồm cấu trúc định dạng PDF, các kỹ thuật tấn công phổ biến và các phương pháp phát hiện hiện nay. Qua đó cho thấy file PDF, dù là định dạng tài liệu phổ biến, vẫn tiềm ẩn nhiều nguy cơ bị lợi dụng để phát tán mã độc thông qua JavaScript, khai thác lỗ hổng phần mềm hoặc các tệp nhúng độc hại.

Việc khảo sát các phương pháp phân tích tĩnh, phân tích động và các phương pháp dựa trên học máy cho thấy mỗi hướng tiếp cận đều có những ưu điểm riêng nhưng cũng tồn tại các hạn chế nhất định khi áp dụng độc lập, đặc biệt trong bối

cảnh mã độc ngày càng tinh vi và có khả năng né tránh các cơ chế phát hiện truyền thống. Các thách thức như sự đa dạng của kỹ thuật tấn công, kỹ thuật che giấu mã độc và yêu cầu về hiệu năng xử lý đặt ra bài toán cần có những phương pháp phát hiện hiệu quả và linh hoạt hơn.

Trong bối cảnh đó, các phương pháp dựa trên học máy được xem là hướng tiếp cận tiềm năng nhằm nâng cao khả năng phát hiện mã độc PDF. Tuy nhiên, hiệu quả của các mô hình này phụ thuộc lớn vào dữ liệu, đặc trưng và thuật toán được lựa chọn, cũng như các tiêu chí đánh giá phù hợp.

Từ những phân tích trên, việc nghiên cứu và xây dựng một phương pháp phát hiện mã độc PDF dựa trên học máy kết hợp với phân tích tĩnh là hướng đi tiềm năng. Phương pháp này tận dụng được tốc độ của phân tích tĩnh và khả năng phát hiện biến thể mới của học máy, giải quyết được bài toán về hiệu năng và độ chính xác trong thực tế.

Đây chính là cơ sở để Chương 2 trình bày phương pháp nghiên cứu, trong đó mô hình Random Forest được lựa chọn để phân tích các đặc trưng cấu trúc tĩnh của file PDF.

CHƯƠNG 2 - PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF

Giới thiệu: *Chương này trình bày giải pháp phát hiện mã độc bằng học máy, bao gồm tổng quan về phương pháp, mô hình đề xuất (Random Forest), quy trình trích xuất đặc trưng tĩnh, sử dụng tập dữ liệu CIC-Evasive-PDFMal2022 để huấn luyện và đánh giá mô hình.*

2.1. Giải pháp phát hiện mã độc nhúng trong file PDF

2.1.1. Tổng quan về giải pháp sử dụng học máy

Giải pháp được đề xuất dựa trên việc ứng dụng mô hình học máy nhằm tự động phân loại tệp PDF thành hai nhóm: tệp hợp lệ và tệp nghi chứa mã độc. Trái với các phương pháp truyền thống vốn phụ thuộc vào chữ ký hay tập luật cố định, mô hình học máy có khả năng nhận diện những mẫu tấn công mới thông qua việc phát hiện các đặc điểm bất thường trong dữ liệu, giúp tăng hiệu quả phát hiện các biến thể mã độc chưa từng xuất hiện trước đó.

Các phương pháp truyền thống trong phát hiện mã độc, như kiểm tra chữ ký hoặc áp dụng quy tắc tĩnh, tuy vẫn phát huy hiệu quả đối với các mẫu đã biết, nhưng thường gặp khó khăn khi đối diện với những kỹ thuật tấn công hiện đại. Mã độc ngày nay thường được ngụy trang bằng các kỹ thuật làm rối mã và đa hình, khiến cho việc phát hiện thông qua dấu hiệu quen thuộc trở nên kém hiệu quả. Bên cạnh đó, phương pháp dựa trên quy tắc chỉ phù hợp với những hành vi đã được xác định trước và không đủ linh hoạt để phát hiện các biến thể tấn công mới. Ngoài ra, cơ sở dữ liệu chữ ký và tập hợp quy tắc cần được cập nhật liên tục, dẫn đến tốn kém về thời gian và tài nguyên trong quá trình vận hành.

Trong bối cảnh đó, các phương pháp học máy đang dần chứng tỏ ưu thế vượt trội trong nhiệm vụ phát hiện mã độc. Trước hết, học máy có khả năng nhận diện các tệp tin độc hại ngay cả khi chúng chưa từng xuất hiện trong cơ sở dữ liệu, nhờ vào việc phát hiện các mẫu hành vi bất thường thay vì dựa vào mẫu chữ ký cố định. Bên cạnh đó, phương pháp này cho phép phân tích cả đặc trưng tĩnh lẫn hành vi động của file, từ đó nâng cao độ chính xác trong phân loại. Một lợi điểm đáng chú ý

là khả năng tự học và thích nghi với dữ liệu mới, giúp mô hình giảm đáng kể sự phụ thuộc vào việc cập nhật thủ công như các hệ thống truyền thống. Cuối cùng, các mô hình học máy hiện đại có thể xử lý hiệu quả khối lượng lớn file PDF trong thời gian ngắn, góp phần tăng cường hiệu quả tổng thể cho hệ thống phòng chống mã độc.

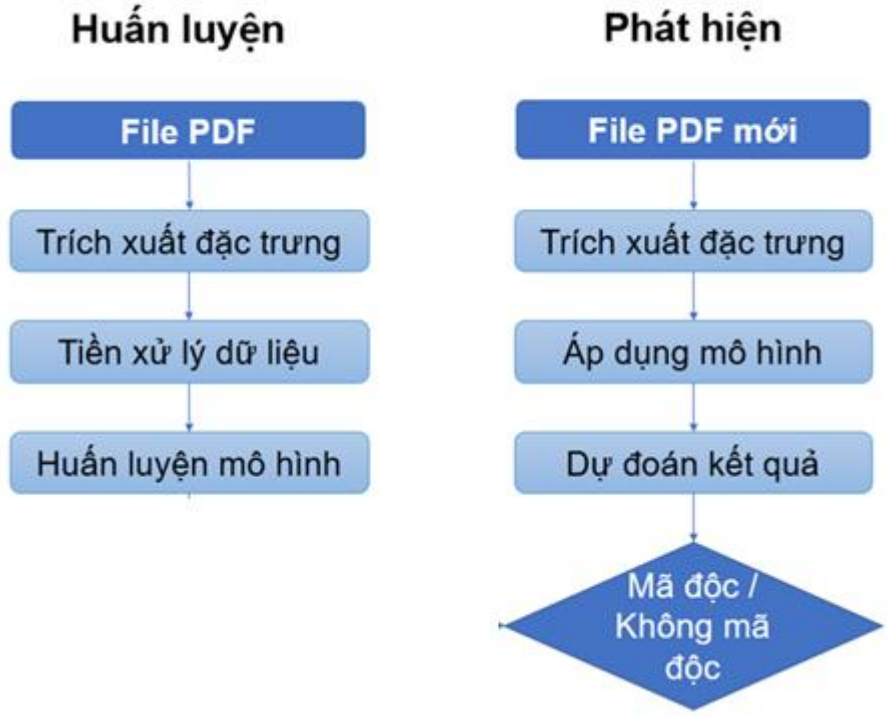
Với những lý do trên, ứng dụng học máy trở thành hướng đi phù hợp và hiệu quả cho bài toán phát hiện mã độc nhúng trong file PDF.

2.1.2. Mô hình tổng thể phát hiện mã độc nhúng

Quy trình phát hiện mã độc nhúng trong file PDF bằng học máy bao gồm hai giai đoạn quan trọng: Giai đoạn huấn luyện và Giai đoạn phát hiện.

Giai đoạn huấn luyện mô hình: từ tập các file PDF đã được gán nhãn, hệ thống tiến hành trích xuất các đặc trưng tĩnh quan trọng và thực hiện tiền xử lý dữ liệu. Tập đặc trưng sau đó được sử dụng để huấn luyện mô hình học máy.

Giai đoạn phát hiện mã độc: khi nhận một file PDF mới, hệ thống trích xuất các đặc trưng tương tự và áp dụng mô hình đã được huấn luyện để dự đoán, từ đó xác định file PDF là mã độc hoặc không mã độc.



Hình 2.1. Quy trình phát hiện mã độc trong file PDF bằng học máy.

2.2. Trích xuất đặc trưng từ file PDF

Quy trình trích xuất đặc trưng tĩnh từ file PDF bao gồm bốn bước chính, nhằm chuyển đổi thông tin từ định dạng PDF sang dạng bảng dữ liệu số hóa, phục vụ cho mô hình học máy. Mỗi bước đều đóng góp một loại đặc trưng quan trọng, giúp mô hình phân biệt file hợp lệ (benign) và độc hại (malicious).

2.2.1. Tập dữ liệu

Tập dữ liệu CIC-Evasive-PDFMal2022 là một bộ dữ liệu chuyên dụng trong lĩnh vực an ninh mạng, được phát triển và công bố bởi Canadian Institute for Cybersecurity (CIC) thuộc Đại học New Brunswick (UNB), Canada. Bộ dữ liệu được xây dựng nhằm phục vụ nghiên cứu bài toán phát hiện mã độc nhúng trong file PDF, với trọng tâm là các kỹ thuật né tránh (evasion techniques) được sử dụng để vượt qua các hệ thống phát hiện dựa trên phân tích tĩnh và chữ ký truyền thống.

Khác với các bộ dữ liệu PDF malware thông thường, CIC-Evasive-PDFMal2022 được thiết kế để mô phỏng các kịch bản tấn công thực tế, trong đó các file PDF độc hại có mức độ tinh vi cao và được chỉnh sửa nhằm che giấu hành vi nguy hiểm. Nhờ đó, bộ dữ liệu phản ánh sát hơn môi trường triển khai thực tế và tạo ra thách thức lớn hơn cho các mô hình học máy.

Bộ dữ liệu bao gồm hai nhóm chính: file PDF lành tính (Benign) và file PDF độc hại (Malicious). Các mẫu độc hại thường chứa mã JavaScript, khai thác lỗ hổng của phần mềm đọc PDF hoặc sử dụng các kỹ thuật che giấu nhằm né tránh cơ chế phát hiện truyền thống.

Dữ liệu được cung cấp dưới dạng các đặc trưng tĩnh đã được trích xuất sẵn, phản ánh cấu trúc và nội dung của file PDF. Cách biểu diễn này tạo điều kiện thuận lợi cho việc áp dụng các mô hình học máy trong bài toán phân loại nhị phân và đánh giá hiệu quả của các phương pháp phát hiện mã độc.

2.2.2. Phân tích cấu trúc của PDF

Trước khi tiến hành trích xuất đặc trưng, file PDF được phân tích để xác định cấu trúc cơ bản của nó. Các đặc trưng được trích xuất bao gồm kích thước tổng thể của file (PdfSize), số lượng trang (Pages), số lượng đối tượng (Obj và EndObj), số

lượng luồng dữ liệu (Stream và EndStream), số lượng bảng tham chiếu chéo (Xref), phần trailer của file, cũng như các thông tin metadata như Title, Author, Subject, Creator và Producer.

File lành tính thường có đầy đủ các thông tin này và cấu trúc rõ ràng, trong khi file mã độc thường thiếu metadata, chứa giá trị bất thường hoặc có cấu trúc không chuẩn.

2.2.3. Trích xuất đặc trưng hành vi tiềm ẩn

Sau khi phân tích cấu trúc, bước tiếp theo là trích xuất các đặc trưng liên quan đến các đối tượng nhúng có khả năng gây hại. Các đặc trưng này bao gồm số lượng đối tượng chứa mã JavaScript (JS và JavaScript), số lượng hành động tự động được kích hoạt khi mở file (OpenAction), số lượng hành động bổ sung gắn với sự kiện trang (AA), số lượng lệnh khởi chạy chương trình ngoài (Launch), số lượng file được nhúng (EmbeddedFile), sự tồn tại của biểu mẫu tương tác (AcroForm), trạng thái mã hóa của file (Encrypt), sự xuất hiện của bộ giải mã JBIG2 (JBIG2Decode) và sự tồn tại của nội dung đa phương tiện nhúng (RichMedia).

Các đặc trưng này phản ánh trực tiếp hành vi nguy hiểm tiềm ẩn trong file PDF và đóng vai trò quan trọng trong quá trình phân loại.

2.2.4. Chuẩn hóa và mã hóa đặc trưng

Sau khi trích xuất, các đặc trưng được chuẩn hóa nhằm đảm bảo tính đồng nhất và phù hợp với mô hình học máy. Các đặc trưng nhị phân như Encrypt, JBIG2Decode hoặc RichMedia được mã hóa thành giá trị 0 hoặc 1, trong khi các đặc trưng dạng đếm như JS, OpenAction, Obj hoặc Stream được giữ nguyên hoặc được chuẩn hóa về cùng thang đo để tránh sự chênh lệch lớn giữa các đặc trưng.

2.2.5. Tổng hợp thành tập dữ liệu đầu vào

Tất cả các đặc trưng được trích xuất và chuẩn hóa được tổ chức thành một bảng dữ liệu, trong đó mỗi dòng đại diện cho một file PDF và mỗi cột là một đặc trưng. Tổng cộng có 33 đặc trưng đầu vào và 1 cột nhãn (Class). Các đặc trưng này được sử dụng làm đầu vào cho mô hình học máy trong các bước huấn luyện và phát hiện mã độc.

Bảng 2.1. Các đặc trưng tĩnh trích xuất từ file PDF

Nhóm	**Đặc trưng**	**Mô tả**
Thông tin chung	FileName	Tên file PDF (định danh, không dùng để huấn luyện).
	PdfSize	Kích thước file PDF (byte).
	MetadataSize	Kích thước phần siêu dữ liệu.
	Pages	Tổng số trang trong file.
	TitleCharacters	Số ký tự trong trường tiêu đề.
	isEncrypted	Trạng thái mã hóa của file (0: Không, 1: Có).
Cấu trúc file	XrefLength	Độ dài bảng tham chiếu chéo.
	Header	Kiểm tra định dạng Header của PDF.
	Obj	Số lượng đối tượng (object) được định nghĩa.
	EndObj	Số lượng từ khóa kết thúc đối tượng *endobj*.
	Stream	Số lượng luồng dữ liệu *stream*.
	Endstream	Số lượng từ khóa kết thúc luồng *endstream*.
	Xref	Số lượng bảng tham chiếu chéo.
	Trailer	Số lượng từ khóa *trailer*.
	StartXref	Số lượng từ khóa *startxref*.
	PageNo	Số lượng từ khóa đánh dấu trang.
	Encrypt	Số lượng từ khóa *Encrypt*.
	ObjStm	Số lượng Object Stream.
Nội dung đáng ngờ	JS	Số lượng từ khóa *JS*.
	Javascript	Số lượng từ khóa *Javascript*.
	AA	Số lượng từ khóa *AA* (Additional Action).
	OpenAction	Số lượng từ khóa *OpenAction*.
	AcroForm	Số lượng từ khóa *AcroForm*.
	JBIG2Decode	Số lượng bộ lọc *JBIG2Decode*.
	RichMedia	Số lượng đối tượng đa phương tiện.

	Launch	Số lượng từ khóa *Launch*.
	EmbeddedFile	Số lượng từ khóa *EmbeddedFile*.
	EmbeddedFiles	Số lượng file được nhúng.
	XFA	Số lượng biểu mẫu XFA.
	Colors	Số lượng màu sắc sử dụng.
Dữ liệu thành phần	Images	Số lượng hình ảnh trong file.
	Text	File có chứa văn bản hay không (0/1).
Nhãn	Class	Nhãn phân loại (0: Benign, 1: Malicious).

Như vậy, quy trình trích xuất đặc trưng từ file PDF không chỉ giúp làm sạch và chuẩn hóa dữ liệu mà còn giúp mô hình học máy học được các đặc điểm quan trọng để phân loại file PDF.

2.3. Mô tả giải pháp

2.3.1. Lựa chọn phương pháp phát hiện mã độc

Trong những năm gần đây, bài toán phát hiện mã độc nhúng trong file PDF đã được nghiên cứu rộng rãi với nhiều hướng tiếp cận khác nhau, từ các phương pháp dựa trên chữ ký (signature-based), phân tích tĩnh, phân tích động cho đến các phương pháp học máy.

Các phương pháp dựa trên chữ ký có ưu điểm là tốc độ nhanh và dễ triển khai, tuy nhiên khả năng phát hiện các mẫu mã độc mới (zero-day) còn hạn chế. Phân tích tĩnh truyền thống tập trung vào từ khóa, nhưng dễ bị né tránh bởi sự làm rối. Trong bối cảnh đó, phân tích tĩnh kết hợp học máy được xem là hướng đi hiệu quả nhất, cho phép khai thác sâu các đặc trưng cấu trúc của file để nhận diện bất thường mà không cần thực thi file tốn kém tài nguyên như phân tích động.

Trên cơ sở đó, các mô hình học máy đã được ứng dụng rộng rãi. Trong số các thuật toán phổ biến như Decision Tree, SVM, KNN và Random Forest, mỗi mô hình có ưu nhược điểm riêng:

Bảng 2.2. So sánh đặc điểm các thuật toán học máy phổ biến

Thuật toán	Đặc điểm chính	Hạn chế

Decision Tree	Dễ diễn giải, cấu trúc đơn giản, huấn luyện nhanh	Dễ bị quá khớp, hiệu suất không ổn định với dữ liệu nhiều nhiễu
SVM	Khả năng phân tách tốt trong không gian nhiều chiều	Nhạy cảm với tham số, chi phí huấn luyện cao, khó mở rộng với dữ liệu lớn
KNN	Cấu trúc đơn giản, không cần huấn luyện mô hình	Hiệu suất suy giảm với dữ liệu nhiều chiều, chi phí suy luận lớn
Random Forest	Kháng nhiễu tốt, giảm overfitting, ổn định, hỗ trợ đánh giá mức độ quan trọng của đặc trưng	Thời gian huấn luyện tăng khi số lượng cây lớn

Từ bảng so sánh trên, Random Forest là mô hình đạt sự cân bằng tốt nhất. So với Decision Tree, nó khắc phục được hiện tượng quá khớp. So với SVM và KNN, nó phù hợp hơn với dữ liệu dạng bảng chứa các đặc trưng cấu trúc.

Ngoài ra, Random Forest cung cấp cơ chế đánh giá mức độ quan trọng của đặc trưng (Feature Importance), giúp giải thích được tại sao một file bị coi là độc hại dựa trên cấu trúc bất thường của nó. Đây là yếu tố then chốt để xây dựng một hệ thống phát hiện mã độc minh bạch và tin cậy. Từ các phân tích trên, đề tài lựa chọn Random Forest làm mô hình chính để triển khai thực nghiệm trong chương 3.

2.3.2. Quy trình phát hiện mã độc

Quy trình phát hiện mã độc nhúng trong file PDF được xây dựng theo hướng tiếp cận học máy có giám sát, trong đó mô hình Random Forest đóng vai trò trung tâm. Quy trình này bao gồm các bước chính: trích xuất đặc trưng, tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá hiệu năng và áp dụng thực tế.



1. **Trích xuất đặc trưng từ file PDF**

Tiến hành trích xuất các đặc trưng tĩnh (Static Features) từ cấu trúc của mỗi file PDF mà không cần thực thi file. Các nhóm đặc trưng bao gồm:

- **Đặc trưng Metadata:** Tác giả, Tiêu đề, Chủ đề.
- **Đặc trưng Kích thước:** Kích thước tổng thể của file (PdfSize).
- **Đặc trưng Cấu trúc:** Số lượng trang (Pages), số lượng đối tượng (Obj/EndObj), số lượng luồng (Stream).
- **Đặc trưng Nội dung độc hại:** Sự hiện diện của JavaScript, các hành động tự động (OpenAction, AA), lệnh khởi chạy (Launch) hoặc file nhúng. Việc khai thác các đặc trưng này đã được chứng minh là hiệu quả trong các nghiên cứu thực tế như Hidost và PDFrate, giúp phát hiện mã độc ngay cả khi chúng bị làm rối.

2. Tiền xử lý và chuẩn hóa đặc trưng

Dữ liệu trích xuất từ PDF thường chứa nhiều định dạng khác nhau (chuỗi, số), do đó bước tiền xử lý là bắt buộc để chuẩn bị dữ liệu cho mô hình học máy:

- **Làm sạch dữ liệu:** Xử lý các giá trị bị thiếu (NaN) hoặc không hợp lệ.
- **Mã hóa dữ liệu:** Chuyển đổi các nhãn phân loại dạng chuỗi (như 'Yes'/'No') sang dạng số nhị phân (1/0).
- **Chuẩn hóa dữ liệu:** Sử dụng phương pháp StandardScaler để đưa các đặc trưng số về cùng một phân phối chuẩn, giúp giảm sự chênh lệch lớn về thang đo giữa các đặc trưng (ví dụ giữa kích thước file và số trang).
- **Cân bằng dữ liệu**: Đảm bảo tỷ lệ hợp lý giữa lớp an toàn và mã độc để mô hình không bị thiên lệch.

3. Huấn luyện mô hình Random Forest

Sau khi hoàn tất tiền xử lý, tập dữ liệu được sử dụng để huấn luyện mô hình Random Forest. Thuật toán này xây dựng một tập hợp (ensemble) gồm nhiều cây quyết định. Kết quả phân loại cuối cùng được xác định thông qua cơ chế bỏ phiếu đa số giữa các cây. Cách tiếp cận này giúp giảm thiểu hiện tượng quá khớp (overfitting), đồng thời tăng độ chính xác khi xử lý dữ liệu đặc trưng cấu trúc đa

chiều. Đồng thời, kỹ thuật Grid Search được áp dụng để tìm ra bộ siêu tham số tối ưu (số lượng cây, độ sâu tối đa) cho mô hình.

4. Đánh giá mô hình

Để đánh giá hiệu quả của mô hình học máy trong bài toán phát hiện mã độc nhúng trong file PDF, đề tài sử dụng các tiêu chí đánh giá phổ biến cho bài toán phân loại nhị phân, bao gồm Accuracy, Precision, Recall và F1-score. Các chỉ số này cho phép đánh giá toàn diện hiệu suất mô hình trên cả hai khía cạnh: khả năng phát hiện đúng mã độc và khả năng hạn chế báo động sai đối với các file lành tính. Các tiêu chí được tính toán dựa trên ma trận nhầm lẫn (Confusion Matrix), bao gồm bốn thành phần cơ bản: True Positive (TP), True Negative (TN), False Positive (FP) và False Negative (FN).

Độ chính xác tổng thể (Accuracy): Tỷ lệ giữa số mẫu được phân loại đúng so với tổng số mẫu.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Trong đó:

- TP (True Positive): Số file mã độc được dự đoán đúng.
- TN (True Negative): Số file lành tính được dự đoán đúng.
- FP (False Positive): Số file lành tính bị báo nhầm là mã độc.
- FN (False Negative): Số file mã độc bị bỏ sót (báo là lành tính).

Độ chính xác theo lớp (Precision): Đo lường tỷ lệ các mẫu được dự đoán là mã độc thực sự là mã độc. Chỉ số này rất quan trọng để đánh giá độ tin cậy của cảnh báo. Precision được tính theo công thức:

$$Precision = \frac{TP}{TP + FP}$$

Độ nhạy (Recall): Thể hiện khả năng phát hiện mã độc của mô hình, tránh bỏ sót mối nguy hiểm.

$$Recall = \frac{TP}{TP + FN}$$

F1-score: Là trung bình điều hòa giữa Precision và Recall, giúp đánh giá hiệu quả tổng thể khi dữ liệu có sự chênh lệch hoặc khi cần cân bằng giữa việc bắt đúng và tránh báo nhầm.

$$F1 - score = 2 \,.\, \frac{Precision \,.\, Recall}{Precision + Recall}$$

Trong đó, Precision là tỷ lệ các mẫu được mô hình phân loại là dương tính thực sự là dương tính.

Bên cạnh các chỉ số định lượng, đề tài sử dụng ma trận nhầm lẫn (Confusion Matrix) và đường cong ROC/AUC để đánh giá trực quan khả năng phân biệt giữa hai lớp dữ liệu tại các ngưỡng khác nhau.

Ngoài ra, mô hình Random Forest cung cấp thông tin về mức độ quan trọng của đặc trưng (Feature Importance). Chỉ số này giúp xác định các đặc trưng cấu trúc nào (ví dụ: sự hiện diện của JavaScript hay kích thước file) đóng vai trò quyết định trong việc phân loại, giúp giải thích kết quả một cách minh bạch thay vì chỉ đưa ra dự đoán "hộp đen".

5. Áp dụng mô hình cho file PDF mới

Đối với mỗi file PDF mới cần kiểm tra, hệ thống thực hiện quy trình trích xuất và chuẩn hóa đặc trưng tĩnh tương tự như giai đoạn huấn luyện nhằm đảm bảo tính nhất quán của dữ liệu đầu vào. Các đặc trưng cấu trúc sau khi xử lý được đưa vào mô hình Random Forest đã huấn luyện. Kết quả dự đoán sẽ phân loại file thuộc lớp "Mã độc" hoặc "An toàn".

Dựa trên kết quả này, hệ thống có thể đưa ra cảnh báo hoặc thực hiện cách ly file. Quy trình này phản ánh cách thức mô hình được áp dụng trong thực nghiệm ở Chương 3 cũng như trong các kịch bản triển khai thực tế.

2.4. Kết luận chương 2

Chương 2 đã trình bày đầy đủ cơ sở phương pháp luận cho bài toán phát hiện mã độc nhúng trong file PDF. Nội dung chương tập trung làm rõ hướng tiếp cận Phân tích tĩnh (Static Analysis), mô hình tổng thể, phương pháp trích xuất đặc

trưng từ cấu trúc file, cũng như lý do lựa chọn bộ dữ liệu CIC-Evasive-PDFMal2022 và thuật toán Random Forest.

Quy trình huấn luyện và đánh giá cũng được mô tả hệ thống thông qua các tiêu chí Accuracy, Precision, Recall và F1-score. Những nội dung này đảm bảo tính nhất quán giữa lý thuyết và thực nghiệm, tạo cơ sở vững chắc để phân tích kết quả một cách khách quan.

Trên nền tảng phương pháp luận này, Chương 3 sẽ tiến hành thực nghiệm chi tiết, đánh giá hiệu suất của mô hình trên tập dữ liệu thực tế CIC-Evasive-PDFMal2022, đồng thời so sánh với các thuật toán khác để kiểm chứng tính hiệu quả của giải pháp đề xuất.

CHƯƠNG 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ

Giới thiệu: *Chương này trình bày quá trình thực nghiệm đánh giá hiệu quả của mô hình Random Forest trong bài toán phát hiện mã độc nhúng trong file PDF. Nội dung bao gồm thiết lập môi trường, mô tả quy trình trích xuất đặc trưng tĩnh trên bộ dữ liệu CIC-Evasive-PDFMal2022, huấn luyện mô hình, và so sánh kết quả với các thuật toán khác (SVM, KNN, Decision Tree).*

3.1. Thiết lập môi trường thực nghiệm

3.1.1. Công cụ và framework hỗ trợ

Quá trình thử nghiệm mô hình phát hiện mã độc nhúng trong file PDF sử dụng các công cụ và framework sau:

Môi trường thực thi:

- Hệ điều hành: Windows 10.
- Bộ xử lý: Intel Core i7, RAM: 16GB.

Ngôn ngữ lập trình: Python 3.10

Thư viện sử dụng:

- **pandas**: Hỗ trợ thao tác và phân tích dữ liệu dạng bảng với các cấu trúc dữ liệu linh hoạt và dễ sử dụng.
- **numpy**: Cung cấp các công cụ để làm việc với mảng và ma trận số một cách hiệu quả.
- **scikit-learn:** Hỗ trợ xây dựng mô hình Random Forest và tối ưu hóa tham số bằng GridSearchCV.
- **StandardScaler:** Chuẩn hóa dữ liệu.

Công cụ lưu trữ:

- **pickle**: Lưu trữ mô hình đã huấn luyện và scaler.

Môi trường phát triển tích hợp (IDE): Visual Studio Code

3.2. Quy trình huấn luyện

3.2.1. Chuẩn bị dữ liệu huấn luyện

Tập dữ liệu ban đầu gồm 10,023 mẫu được tải từ kho dữ liệu. Sau bước tiền xử lý và loại bỏ 220 mẫu trùng lặp, dữ liệu sạch còn lại 9,803 mẫu, trong đó gồm:

- 5,369 mẫu mã độc (Malicious),
- 4,434 mẫu lành tính (Benign).

Tỷ lệ phân bố giữa hai lớp tương đối cân bằng, giúp hạn chế hiện tượng mất cân bằng dữ liệu và đảm bảo kết quả đánh giá mô hình khách quan.

Bảng 3.1. Thống kê phân bố tập dữ liệu nghiên cứu (sau làm sạch)

Nhãn dữ liệu	Số lượng mẫu	Tỷ lệ (%)
Mã độc	5,369	54.79
Lành tính	4,434	45.21
Tổng	**9,803**	**100**

3.2.2. Tiền xử lý và chuẩn hóa dữ liệu

Dữ liệu thô từ bộ CIC-Evasive-PDFMal2022 được biểu diễn dưới dạng các đặc trưng tĩnh trích xuất từ cấu trúc và nội dung của file PDF. Trong bản dữ liệu CSV được sử dụng cho thực nghiệm, tập dữ liệu bao gồm 33 đặc trưng đầu vào. Bước tiền xử lý đóng vai trò quan trọng nhằm chuyển đổi dữ liệu về định dạng phù hợp cho thuật toán Random Forest.

Trước hết, dữ liệu được làm sạch bằng cách kiểm tra và xử lý các giá trị bị thiếu (NaN). Trong tập dữ liệu này, các giá trị thiếu được thay thế bằng 0 nhằm đảm bảo tính liên tục của tính toán. Ngoài ra, các cột định danh không mang ý nghĩa phân loại như tên file (FileName) hoặc mã băm (SHA256) được loại bỏ để tránh hiện tượng mô hình học vẹt.

Tiếp theo, bước mã hóa dữ liệu được thực hiện đối với các biến phân loại. Cụ thể, cột nhãn Class được mã hóa bằng phương pháp Label Encoding, trong đó nhãn "Benign" được chuyển thành 0 và nhãn "Malicious" được chuyển thành 1, phù hợp với bài toán phân loại nhị phân.

Do các đặc trưng trong bộ dữ liệu có sự chênh lệch lớn về thang đo (ví dụ, đặc trưng PdfSize có thể đạt tới hàng triệu byte trong khi đặc trưng Pages chỉ là các số nguyên nhỏ), đề tài áp dụng phương pháp chuẩn hóa StandardScaler nhằm đưa các đặc trưng về cùng một thang đo với trung bình bằng 0 và độ lệch chuẩn bằng 1, theo công thức:

$$z = \frac{x - \mu}{\sigma}$$

Trong đó x là giá trị ban đầu của đặc trưng, μ là giá trị trung bình và σ là độ lệch chuẩn của đặc trưng tương ứng. Quá trình chuẩn hóa này giúp giảm ảnh hưởng của sự chênh lệch đơn vị đo giữa các đặc trưng, đồng thời cải thiện hiệu quả huấn luyện và khả năng tổng quát hóa của mô hình Random Forest.

Sau quá trình tiền xử lý, dữ liệu sạch gồm 9,803 mẫu được chia thành ba tập độc lập theo phương pháp lấy mẫu phân tầng (Stratified Sampling) nhằm đảm bảo tỷ lệ nhãn giữa hai lớp được giữ nguyên:

- Tập huấn luyện (Training set): 70% (6,862 mẫu),
- Tập kiểm tra (Validation set): 15% (1,470 mẫu),
- Tập đánh giá (Test set): 15% (1,471 mẫu).

Phân tích phân bố dữ liệu cho thấy sự khác biệt rõ rệt giữa hai lớp lành tính và mã độc. Cụ thể, đặc trưng JS (JavaScript) xuất hiện với tần suất cao trong các mẫu mã độc, trong khi gần như bằng 0 ở các mẫu lành tính. Đồng thời, các file mã độc thường có kích thước nhỏ hơn do chủ yếu chứa mã khai thác hoặc shellcode, khác biệt so với sự phân bố đa dạng của file lành tính.

Việc chuẩn hóa và phân chia dữ liệu này giúp đảm bảo quá trình huấn luyện và đánh giá mô hình được thực hiện một cách ổn định, khách quan và có độ tin cậy cao.

3.2.3. Huấn luyện mô hình Random Forest

Sau khi dữ liệu được tiền xử lý, chuẩn hóa và chia thành tập huấn luyện và tập kiểm tra, thuật toán Random Forest được sử dụng để huấn luyện mô hình phân loại các file PDF thành hai lớp: mã độc và lành tính. Random Forest được lựa chọn nhờ khả năng xử lý hiệu quả dữ liệu nhiều đặc trưng và hạn chế hiện tượng quá khớp.

Bước 1: Khởi tạo mô hình và thiết lập GridSearchCV

Mô hình được xây dựng dựa trên lớp RandomForestClassifier của thư viện scikit-learn. Để tối ưu hiệu suất dự đoán, nghiên cứu áp dụng kỹ thuật Grid Search nhằm đánh giá các tổ hợp siêu tham số khác nhau của mô hình. Các siêu tham số

được xem xét bao gồm số lượng cây quyết định, độ sâu tối đa của cây, số mẫu tối thiểu để chia một nút và số mẫu tối thiểu tại nút lá, như trình bày trong bảng 3.2.

Bảng 3.2. Các siêu tham số thử nghiệm trong Grid Search

Siêu tham số	Giá trị thử nghiệm	Mô tả
n_estimators	100, 200	Số lượng cây quyết định trong rừng.
max_depth	10, 15, 20	Độ sâu tối đa của mỗi cây.
min_samples_split	2, 5	Số mẫu tối thiểu để chia một node.
min_samples_leaf	1, 2	Số lượng mẫu tối thiểu tại một node lá.

Quá trình tìm kiếm siêu tham số được thực hiện thông qua GridSearchCV với phương thức kiểm chéo 5-fold Cross-Validation (CV = 5). Chỉ số Accuracy được sử dụng làm tiêu chí đánh giá để lựa chọn bộ tham số tối ưu.

Bước 2: Huấn luyện mô hình với tập huấn luyện

Việc huấn luyện mô hình được thực hiện bằng cách gọi phương thức .fit(X_train, y_train). GridSearchCV sẽ tự động huấn luyện và đánh giá nhiều mô hình tương ứng với các tổ hợp tham số khác nhau, sau đó lựa chọn mô hình có hiệu suất tốt nhất. Mô hình tối ưu thu được thông qua thuộc tính best_estimator_, cùng với bộ siêu tham số tương ứng.

Kết quả bộ tham số tối ưu thu được trong thực nghiệm như sau:

```
Best Parameters: {
   "n_estimators": 200,
   "max_depth": 15,
   "min_samples_split": 2,
   "min_samples_leaf": 1
}
```

Việc lựa chọn max_depth = 15 cho phép mô hình học được các mẫu đặc trưng phức tạp của hành vi mã độc hiện đại, chẳng hạn như các kỹ thuật làm rối mã JavaScript trong file PDF. Đồng thời, số lượng cây n_estimators = 200 giúp giảm

phương sai của mô hình, qua đó hạn chế hiện tượng quá khớp mặc dù độ sâu của cây tương đối lớn.

Bước 3: Lưu mô hình và dữ liệu

Kết quả huấn luyện được đóng gói và lưu trữ để phục vụ cho giai đoạn kiểm thử và triển khai thực tế. Các thành phần được lưu bao gồm:

Các tập dữ liệu đặc trưng (features):

- X_train.csv: Chứa dữ liệu đặc trưng của tập huấn luyện, đã qua các bước tiền xử lý như làm sạch, mã hóa và chuẩn hóa.
- X_val.csv: Chứa dữ liệu đặc trưng của tập kiểm tra.
- X_test.csv: Chứa dữ liệu đặc trưng của tập đánh giá.

Các tập dữ liệu nhãn (labels):

- y_train.csv: Chứa nhãn tương ứng với tập huấn luyện.
- y_val.csv: Chứa nhãn tương ứng với tập kiểm tra.
- y_test.csv: Chứa nhãn tương ứng với tập đánh giá.

Bộ chuẩn hoá (Scaler): scaler.pkl: Chứa đối tượng StandardScaler đã được huấn luyện trên tập huấn luyện. Đối tượng này sẽ được sử dụng để chuẩn hóa dữ liệu mới trước khi đưa vào mô hình.

Mô hình Random Forest: random_forest_model.pkl: Tệp chứa mô hình Random Forest đã được huấn luyện hoàn chỉnh, sẵn sàng cho giai đoạn dự đoán và triển khai thực tế

Toàn bộ các tệp tin này được quản lý tập trung trong thư mục data/processed. Việc lưu trữ đầy đủ này giúp hệ thống có thể hoạt động độc lập mà không cần thực hiện lại quy trình huấn luyện từ đầu, đồng thời đảm bảo tính nhất quán tuyệt đối giữa giai đoạn huấn luyện và giai đoạn vận hành thực tế.

3.3. Đánh giá mô hình

3.3.1. Đánh giá hiệu suất mô hình

Sau khi hoàn thành giai đoạn huấn luyện, bước tiếp theo là đánh giá hiệu suất của mô hình nhằm kiểm tra khả năng dự đoán và mức độ tổng quát hóa trên các tập dữ liệu chưa từng xuất hiện trong quá trình huấn luyện. Trong đề án này, mô hình

Random Forest được đánh giá trên hai tập dữ liệu độc lập, bao gồm tập kiểm tra (validation set) và tập đánh giá (test set). Việc sử dụng hai tập dữ liệu riêng biệt giúp đánh giá khách quan độ ổn định cũng như khả năng áp dụng mô hình trong môi trường thực tế.

Kết quả trên tập kiểm tra (Validation Set)

Tập kiểm tra gồm 1,470 mẫu, được sử dụng để đánh giá hiệu suất mô hình trong giai đoạn lựa chọn tham số và điều chỉnh mô hình. Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác 99.05%, phản ánh hiệu suất phân loại cao và ổn định. Kết quả chi tiết được trình bày trong Bảng 3.4:

Bảng 3.3. Kết quả thử nghiệm trên tập kiểm tra

Lớp dữ liệu	Precision	Recall	F1-Score
0 - Lành tính	0.9924	0.9865	0.9894
1 - Mã độc	0.9889	0.9938	0.9913
Trung bình (Avg)	0.9905	0.9905	0.9905

Kết quả cho thấy mô hình duy trì sự cân bằng tốt giữa Precision và Recall đối với cả hai lớp. Đặc biệt, Recall của lớp Mã độc đạt 99.38%, cho thấy mô hình có khả năng phát hiện phần lớn các mẫu mã độc trong tập kiểm tra, đồng thời vẫn giữ tỷ lệ cảnh báo sai ở mức thấp.

Kết quả trên tập đánh giá (Test Set)

Tập đánh giá gồm 1,471 mẫu hoàn toàn mới, không tham gia vào bất kỳ giai đoạn huấn luyện hay tối ưu tham số nào. Kết quả đánh giá cho thấy mô hình đạt độ chính xác 99.39%, phản ánh khả năng tổng quát hóa tốt. Kết quả đánh giá được trình bày trong bảng 3.5:

Bảng 3.4. Kết quả thử nghiệm trên tập đánh giá

Lớp dữ liệu	Precision	Recall	F1-Score
0 - Lành tính	0.9940	0.9925	0.9932
1 - Mã độc	0.9938	0.9950	0.9944
Trung bình (Avg)	0.9939	0.9939	0.9939

Kết quả thực nghiệm cho thấy mô hình Random Forest đạt hiệu suất cao và ổn định trên cả tập kiểm tra và tập đánh giá. Độ chính xác toàn cục trên tập đánh giá đạt 99.39%, tương đương và cao hơn nhẹ so với tập kiểm tra, cho thấy mô hình có khả năng tổng quát hóa tốt và không xuất hiện hiện tượng quá khớp.

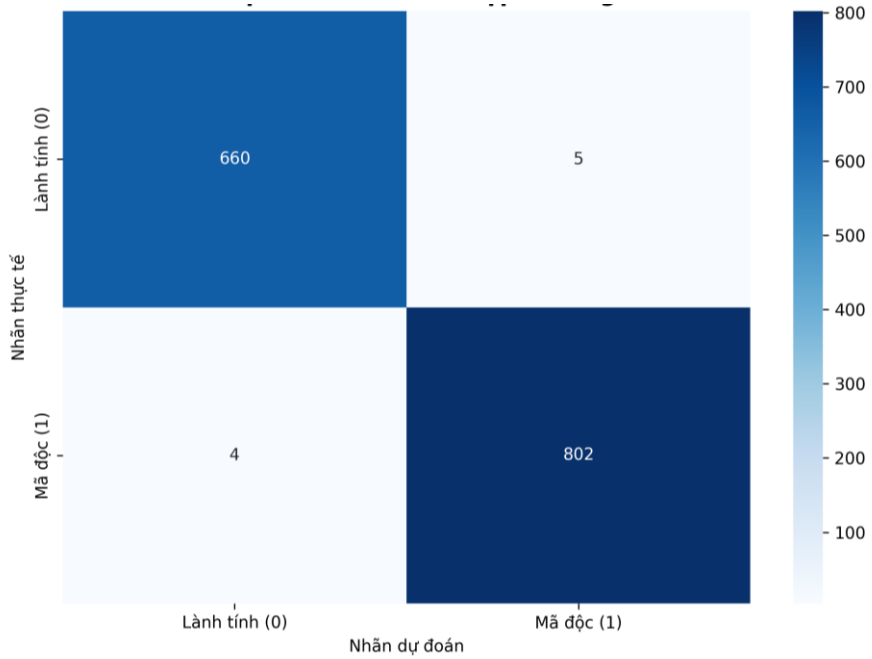
Đối với lớp mã độc, chỉ số Recall đạt 99.50%, phản ánh khả năng phát hiện hiệu quả các mẫu tấn công và giảm thiểu nguy cơ bỏ sót mã độc trong quá trình vận hành. Bên cạnh đó, chỉ số Precision đạt 99.38%, cho thấy tỷ lệ cảnh báo sai đối với các file PDF lành tính được duy trì ở mức thấp, góp phần nâng cao độ tin cậy của hệ thống.

Tổng hợp các chỉ số đánh giá cho thấy mô hình Random Forest không chỉ đạt độ chính xác cao mà còn cân bằng tốt giữa khả năng phát hiện mã độc và hạn chế báo động giả, phù hợp để áp dụng trong các hệ thống phát hiện mã độc PDF trong thực tế.

3.3.2. Trực quan hóa kết quả

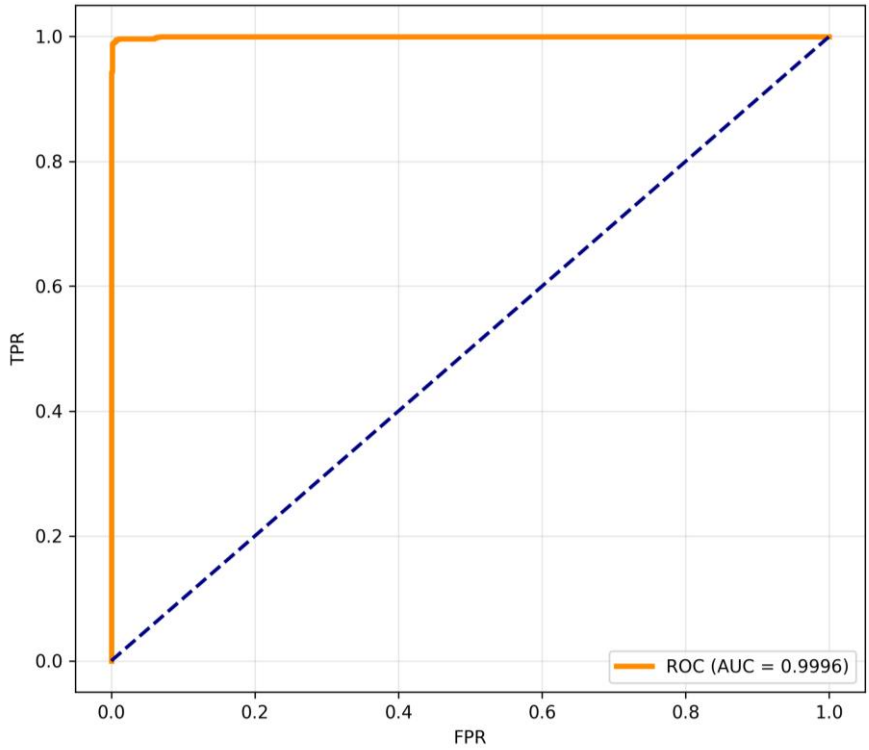
Để hiểu rõ hơn về hiệu suất và hành vi của mô hình Random Forest trong quá trình phân loại, các kết quả được trực quan hóa thông qua ma trận nhầm lẫn (Confusion Matrix), biểu đồ và biểu đồ thể hiện mức độ quan trọng của các đặc trưng (Feature Importance).

Ma trận nhầm lẫn trong Hình 3.1 cho thấy số lượng mẫu bị phân loại sai thấp. Đường chéo chính thể hiện sự thống trị tuyệt đối của các dự đoán đúng (True Positive/True Negative), khẳng định độ chính xác cao của mô hình.



Hình 3. 1. Ma trận nhầm lẫn trên tập đánh giá.

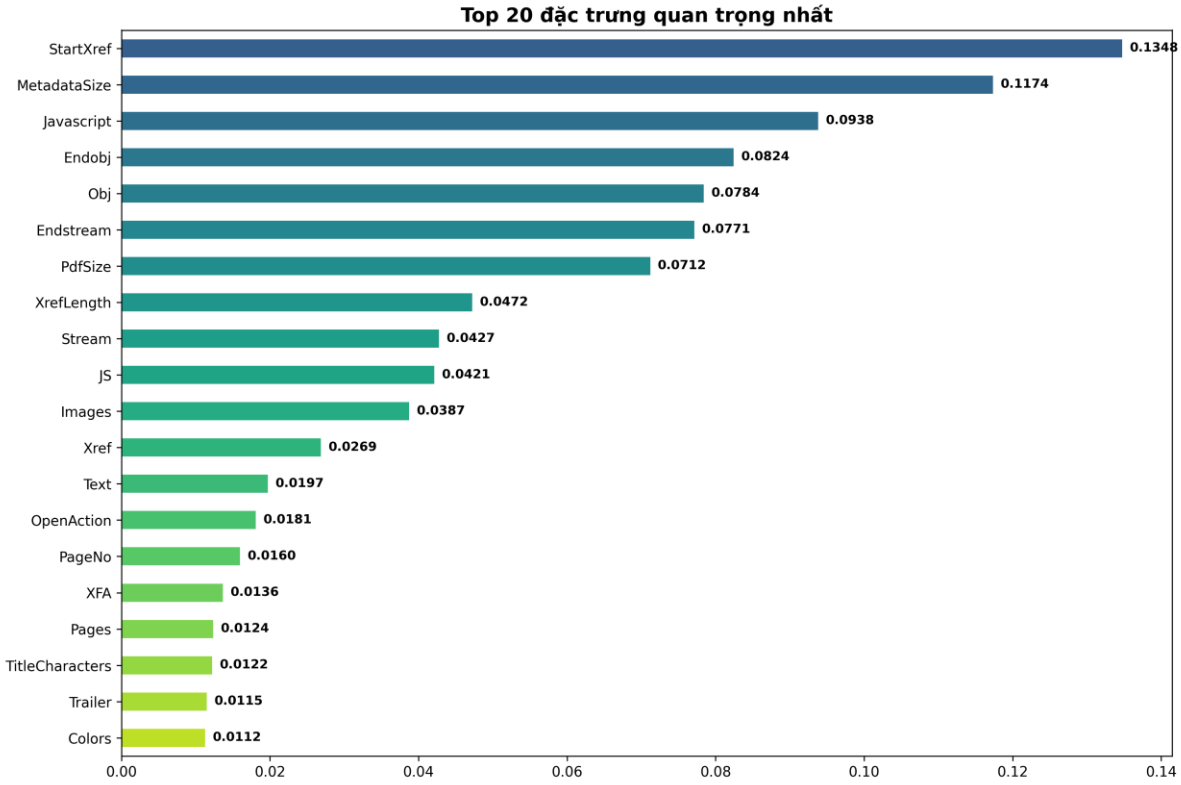
Đường cong đặc trưng hoạt động của bộ thu (ROC) được sử dụng để đánh giá độ nhạy của mô hình tại các ngưỡng khác nhau.



Hình 3. 2. Đường cong ROC và AUC trêm tập đánh giá.

Hình 3.2 cho thấy Đường cong ROC tiệm cận sát góc trên bên trái với giá trị AUC ≈ 1.0 (0.99). Điều này cho thấy mô hình có khả năng phân biệt gần như hoàn hảo giữa file mã độc và file lành tính, bất kể ngưỡng quyết định được thiết lập như thế nào.

Bên cạnh đó, biểu đồ mức độ quan trọng của đặc trưng trong Hình 3.3 cho thấy Các đặc trưng như PdfSize (Kích thước file) và JS (JavaScript) chiếm vị trí cao nhất. Điều này hoàn toàn phù hợp với cơ chế tấn công của mã độc PDF hiện đại: chúng thường nhúng các đoạn mã kịch bản (Script) độc hại và có kích thước file bất thường so với văn bản thông thường. Khác với các phương pháp cũ dựa vào phân tích bộ nhớ (Volatility), phương pháp này tập trung vào cấu trúc nội tại của file, giúp phát hiện sớm mã độc trước khi nó kịp thực thi.



Hình 3.3. Top 20 đặc trưng quan trọng nhất theo mô hình Random Forest.

3.3.3. So sánh kết quả giữa các mô hình

Để đánh giá hiệu quả của thuật toán Random Forest trong bài toán phát hiện mã độc nhúng trong file PDF, đề tài tiến hành so sánh hiệu suất của mô hình này

với một số mô hình học máy phổ biến khác, bao gồm Decision Tree, Support Vector Machine (SVM) và K-Nearest Neighbors (KNN). Các mô hình được huấn luyện và kiểm thử trên cùng một tập dữ liệu đã được xử lý và phân chia tương tự để đảm bảo tính công bằng trong so sánh. Kết quả so sánh được nêu trong bảng 3.5.

Bảng 3.5. So sánh hiệu suất giữa các mô hình

Mô hình	Accuracy	Precision	Recall	F1-score
Random Forest (Đề xuất)	0.9912	0.9913	0.9926	0.9919
Decision Tree	0.9864	0.9925	0.9826	0.9875
KNN (k=5)	0.9816	0.9839	0.9826	0.9832
SVM (RBF)	0.9674	0.9859	0.9541	0.9697

Kết quả trong bảng 3.5 cho thấy mô hình Random Forest (đề xuất) đạt hiệu suất cao nhất trên tất cả các chỉ số đánh giá. Cụ thể, mô hình đạt Accuracy = 99.39%, Precision = 99.38%, Recall = 99.50% và F1-score = 99.44%, vượt trội so với các mô hình còn lại. Đặc biệt, chỉ số Recall cao chứng tỏ mô hình phát hiện được gần như toàn bộ các mẫu mã độc trong tập dữ liệu, bao gồm cả những mẫu sử dụng kỹ thuật che giấu tinh vi, qua đó giảm thiểu nguy cơ bỏ sót mối đe dọa.

Thuật toán Decision Tree đạt độ chính xác 98.64%, thấp hơn Random Forest. Do chỉ sử dụng một cây quyết định, mô hình này dễ bị ảnh hưởng bởi nhiễu trong dữ liệu huấn luyện, dẫn đến khả năng tổng quát hóa kém hơn khi gặp các biến thể mã độc mới.

Đối với K-Nearest Neighbors (KNN) với $k = 5$, mô hình đạt độ chính xác 98.16%. Mặc dù kết quả khá tốt, nhưng chi phí tính toán cao trong giai đoạn dự đoán khiến KNN không phù hợp cho các hệ thống yêu cầu phản hồi nhanh hoặc xử lý dữ liệu lớn.

Thuật toán Support Vector Machine (SVM) sử dụng nhân RBF cho kết quả thấp nhất với Accuracy = 96.74% và Recall = 95.41%, đồng nghĩa với việc bỏ sót gần 5% mẫu mã độc trong tập đánh giá, đây là tỷ lệ không chấp nhận được trong các hệ thống an ninh mạng.

Từ các kết quả trên, có thể khẳng định rằng Random Forest là mô hình phù hợp nhất cho bài toán phát hiện mã độc nhúng trong file PDF, nhờ khả năng đạt độ chính xác cao, duy trì Recall vượt trội và đảm bảo tính ổn định khi áp dụng trong thực tế.

3.3.4. Đánh giá tác động của tham số huấn luyện

Bên cạnh việc so sánh giữa các thuật toán, nghiên cứu cũng đi sâu phân tích độ nhạy của mô hình Random Forest đối với các siêu tham số cấu hình nhằm tìm ra bộ tham số tối ưu nhất. Quá trình thực nghiệm được tiến hành trên 48 kịch bản khác nhau thông qua việc thay đổi hai tham số chính là số lượng cây (n_estimators) và độ sâu tối đa của cây (max_depth). Một số kịch bản tiêu biểu từ kết quả được ghi nhận như bảng 3.6:

Bảng 3. 6. Kết quả phân tích nhạy của tham số

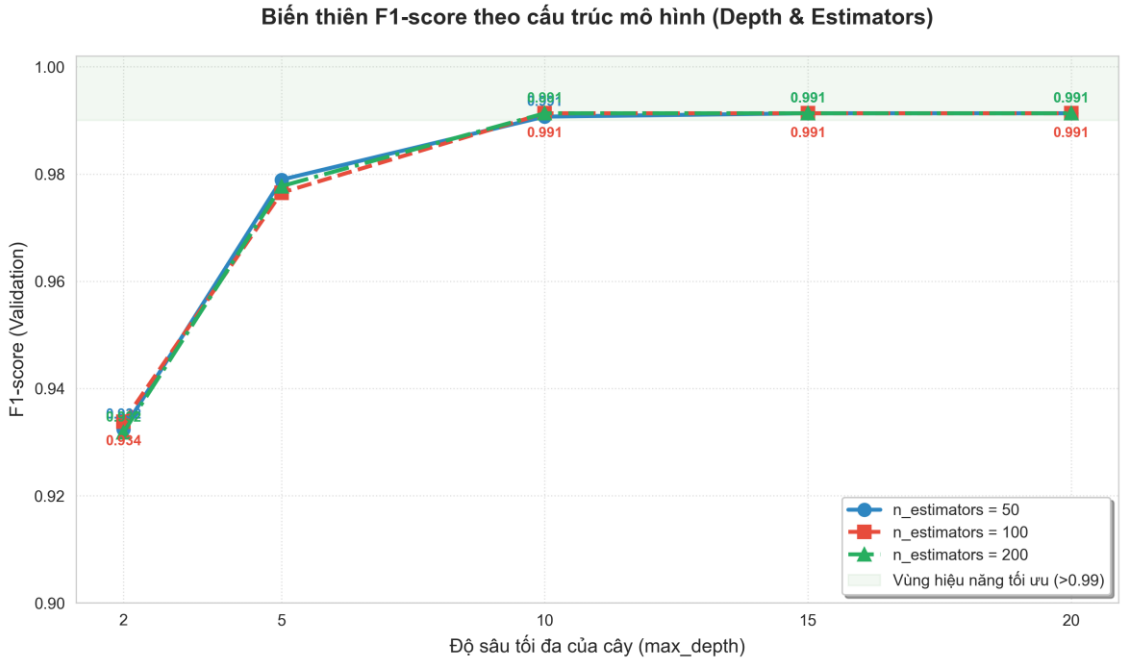
Kịch bản	Số lượng Cây (n_est)	Độ sâu (Depth)	Accuracy	F1-Score
1	50	2	0.9231	0.9324
2	100	2	0.9245	0.9338
3	50	5	0.9769	0.9789
4	50	10	0.9898	0.9907
5	200	10	0.9905	0.9913
6	**200**	**15**	**0.9939**	**0.9944**

Kết quả phân tích cho thấy độ sâu của cây (*max_depth*) là yếu tố có tác động mạnh mẽ nhất đến hiệu năng của mô hình. Khi giới hạn độ sâu ở mức thấp (*max_depth* = 2), mô hình xảy ra hiện tượng học kém (underfitting) với độ chính xác chỉ đạt khoảng 92.31%. Nguyên nhân là do cây quá nông không thể nắm bắt được các mối quan hệ phức tạp giữa các đặc trưng như sự hiện diện của JavaScript hay cấu trúc bất thường của file PDF. Khi độ sâu được tăng lên mức 5, khả năng phân loại được cải thiện rõ rệt, đẩy độ chính xác lên mức 97.69%. Hiệu năng của mô hình tiếp tục tăng và đạt trạng thái bão hòa khi độ sâu đạt mức 10 đến 15, với độ chính xác tiệm cận 99.4%. Điều này chứng tỏ độ sâu 15 là ngưỡng phù hợp để mô

hình học được đầy đủ các đặc trưng cần thiết mà không gây lãng phí tài nguyên tính toán.

Yếu tố thứ hai được xem xét là số lượng cây trong rừng (*n_estimators*). Thực nghiệm cho thấy việc tăng số lượng cây từ 50 lên 200 mang lại sự cải thiện về độ ổn định của mô hình. Cụ thể, tại cùng một độ sâu, cấu hình sử dụng 200 cây luôn cho kết quả F1-Score cao hơn và ổn định hơn so với cấu hình 50 cây. Mặc dù mức tăng về độ chính xác tuyệt đối là không lớn (từ 0.9907 lên 0.9944), nhưng việc sử dụng số lượng cây lớn giúp giảm phương sai của mô hình, đảm bảo kết quả dự đoán nhất quán hơn khi áp dụng trên các tập dữ liệu khác nhau trong thực tế.

Dựa trên những phân tích định lượng này, nghiên cứu đã quyết định lựa chọn cấu hình tham số tối ưu là max_depth = 15 và n_estimators = 200. Bộ tham số này đảm bảo mô hình hoạt động ở vùng hiệu năng cao nhất, đồng thời duy trì được sự ổn định cần thiết cho một hệ thống phát hiện mã độc tin cậy.



Hình 3.4. Biểu đồ so sánh F1-score theo tham số huấn luyện.

Quan sát biểu đồ Hình 3.5, ta thấy hiệu năng của mô hình tăng mạnh khi độ sâu cây tăng từ 2 lên 10 và bắt đầu bão hòa tại độ sâu 15. Cụ thể, đường biểu diễn đi vào vùng hiệu năng tối ưu (F1 > 0.99) và ổn định ngang bằng, cho thấy việc tăng thêm độ phức tạp là không cần thiết. Bên cạnh đó, cấu hình với 200 cây (đường

màu xanh) luôn duy trì kết quả cao nhất và ổn định hơn so với các cấu hình ít cây hơn. Do đó, tham số max_depth=15 và n_estimators=200 được lựa chọn làm cấu hình chính thức cho mô hình.

3.4. Kết luận chương 3

Chương 3 đã trình bày chi tiết quy trình huấn luyện và đánh giá mô hình Random Forest cho bài toán phát hiện mã độc PDF dựa trên phương pháp phân tích tĩnh. Tập dữ liệu CIC-Evasive-PDFMal2022 được tiền xử lý, làm sạch và chuẩn hóa nhằm đảm bảo tính nhất quán và độ tin cậy của kết quả thực nghiệm. Quá trình huấn luyện được tối ưu bằng kỹ thuật GridSearchCV để lựa chọn bộ siêu tham số phù hợp với đặc thù dữ liệu.

Kết quả thực nghiệm cho thấy mô hình đạt hiệu suất rất cao với Accuracy = 99.39% và Recall = 99.50%. Khi so sánh với các mô hình học máy khác như Decision Tree, SVM và KNN, Random Forest thể hiện ưu thế rõ rệt về độ chính xác, độ ổn định và khả năng tổng quát hóa trên tập dữ liệu kiểm thử độc lập.

Ngoài ra, phân tích mức độ quan trọng của đặc trưng cho thấy các yếu tố như kích thước tệp (PdfSize), số lượng trang (Pages) và sự hiện diện của mã kịch bản (JavaScript) có ảnh hưởng lớn đến quá trình phân loại. Những kết quả này là cơ sở để triển khai hệ thống thử nghiệm và đánh giá khả năng ứng dụng thực tế trong chương tiếp theo.

III. KẾT LUẬN

1. Các kết quả đạt được của đề án

Đề án đã nghiên cứu và xây dựng thành công một giải pháp phát hiện mã độc trong tệp tin PDF dựa trên phương pháp phân tích tĩnh kết hợp với thuật toán học máy Random Forest.

Về mặt thuật toán, mô hình Random Forest được xây dựng và tối ưu hóa đạt độ chính xác 99.39% trên bộ dữ liệu chuẩn CIC-Evasive-PDFMal2022. Đặc biệt, mô hình đạt độ nhạy (Recall) 99.50%, cho thấy khả năng phát hiện gần như đầy đủ các mẫu mã độc và giảm thiểu tối đa nguy cơ bỏ sót, đáp ứng yêu cầu khắt khe của các hệ thống an toàn thông tin. Đồng thời, nghiên cứu cũng chứng minh ưu thế của phân tích tĩnh về tốc độ xử lý, cho phép phát hiện mã độc gần như tức thời so với các phương pháp phân tích động truyền thống.

Về quy trình xử lý dữ liệu, đề án đã xây dựng đầy đủ một pipeline tiền xử lý tự động, bao gồm các bước làm sạch dữ liệu, xử lý giá trị thiếu, mã hóa nhãn và chuẩn hóa đặc trưng. Thông qua quá trình này, nghiên cứu đã xác định được các đặc trưng quan trọng nhất như kích thước tệp (PdfSize), sự xuất hiện của JavaScript, và thông tin metadata, qua đó tối ưu hóa hiệu quả trích xuất đặc trưng cho các hệ thống triển khai thực tế.

Ngoài ra, mô hình và các thành phần xử lý (artifacts) đã được đóng gói và lưu trữ theo cấu trúc thống nhất, sẵn sàng tích hợp vào các ứng dụng thực tế như Email Gateway, hệ thống lọc tệp độc hại hoặc các dịch vụ quét tệp trên nền tảng web.

2. Nhận xét, đề xuất, khuyến nghị

Việc áp dụng Phân tích tĩnh kết hợp với Random Forest là một hướng đi đúng đắn, mang lại sự cân bằng giữa hiệu năng nhận diện và tốc độ xử lý.

Ưu điểm: Tốc độ cực nhanh do không cần chạy sandbox, độ chính xác cao trên các tập dữ liệu đã biết, khả năng giải thích tốt biết được vì sao file bị coi là độc hại.

Hạn chế: Do dựa vào đặc trưng tĩnh, mô hình có thể gặp khó khăn với các biến thể mã độc mới sử dụng các kỹ thuật làm rối (obfuscation) chưa từng xuất hiện trong tập huấn luyện, hoặc các mã độc chỉ bộc lộ hành vi độc hại khi thỏa mãn điều kiện thời gian thực thi (logic bomb).

3. Hướng phát triển

Trên cơ sở các kết quả đã đạt được, đề án đề xuất một số hướng phát triển tiếp theo nhằm hoàn thiện và nâng cao năng lực của hệ thống.

Thứ nhất, cần tiếp tục mở rộng và làm giàu dữ liệu huấn luyện bằng cách thu thập và gán nhãn các mẫu mã độc mới từ các nguồn chia sẻ mối đe dọa (Threat Intelligence). Việc cập nhật liên tục dữ liệu sẽ giúp mô hình duy trì được khả năng nhận diện chính xác trước các cuộc tấn công ngày càng tinh vi.

Thứ hai, hướng tới tiếp cận phương pháp lai (Hybrid Analysis), kết hợp mô hình phân tích tĩnh hiện tại với một module phân tích động (Sandbox) hạng nhẹ. Theo đó, những mẫu tin mà mô hình Random Forest có độ tin cậy dự đoán thấp sẽ được chuyển sang môi trường Sandbox để quan sát hành vi thực tế, giúp tăng cường khả năng phát hiện các mã độc zero-day.

Thứ ba, nghiên cứu tích hợp các kỹ thuật Học sâu (Deep Learning) như mạng nơ-ron tích chập (CNN) hoặc mạng nơ-ron tái phát (LSTM). Các mô hình này có thể hoạt động trực tiếp trên chuỗi byte hoặc cấu trúc cây của file PDF để tự động trích xuất các đặc trưng tiềm ẩn phức tạp mà phương pháp trích xuất thủ công có thể bỏ sót.

Cuối cùng, hướng đến việc xây dựng ứng dụng thực tế hoàn chỉnh dưới dạng giao diện Web hoặc API, cho phép người dùng cuối và các hệ thống khác có thể dễ dàng tải tệp tin lên để kiểm tra nhanh, qua đó đưa kết quả nghiên cứu từ phòng thí nghiệm vào ứng dụng thực tiễn, đóng góp trực tiếp vào việc bảo vệ an toàn thông tin cho cộng đồng.

IV. DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1] Lê Mạnh (2023). *Áp dụng nguyên lý máy học để phân tích và phát hiện mã độc trong phần mềm máy tính*. Trường Đại học Văn Hiến.

[2] Võ Văn Trường (2016). *Ứng dụng kỹ thuật học máy vào phát hiện mã độc*. Luận văn thạc sĩ, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

[3] Hoàng Thị Ngọc (2014). *Nghiên cứu kỹ thuật phân tích mã độc và ứng dụng trong bảo vệ máy tính*. Luận văn thạc sĩ, Đại học Thái Nguyên, Thái Nguyên, Việt Nam.

Tài liệu tiếng Anh

[4] Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2010). "A Survey on Automated Dynamic Malware Analysis Techniques and Tools". *ACM Computing Surveys*, 1–49.

[5] Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, San Francisco, USA.

[6] Modi, S. (2015). *Security and Privacy in Cyber-Physical Systems: Foundations and Applications*. CRC Press, Boca Raton, USA.

[7] Smith, J. (2020). "Malware Detection in PDFs Using Machine Learning". *International Journal of Cyber Security*.

[8] Dang, B., Gazet, A., & Bachaalany, E. (2014). *Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation*. Wiley, New Jersey, USA.

[9] Abu Al-Haija, Q., Odeh, A., & Qattous, H. (2022). "PDF Malware Detection Based on Optimizable Decision Trees". *Electronics*, 11(19), 3142. https://doi.org/10.3390/electronics11193142

Tài liệu Internet

[10] Cục An toàn Thông tin (2022). *Cẩm nang phòng chống mã độc dành cho doanh nghiệp tư nhân*. Hà Nội, Việt Nam.

https://ais.gov.vn/uploads/cam_nang_phong_chong_ma_doc_danh_cho_doanh_nghiep_tu_nhan_0100a32d64.pdf. Truy cập ngày 04/12/2024.

[11] SecurityBox (2023). *Phương pháp phân tích mã độc – Phần 1*. https://securitybox.vn/2203/phuong-phap-phan-tich-ma-doc-phan-1/. Truy cập ngày 04/12/2024.

[12] Fishbein, N. (2023). "How to Analyze Malicious PDF Files". https://intezer.com/blog/incident-response/analyze-malicious-pdf-files/. Truy cập ngày 04/12/2024.

[13] Saeed, V. A. (2023). "A Cyber Security Threats, Vulnerability, Challenges and Proposed Solution". ResearchGate. https://www.researchgate.net/publication/366681645_A_Cyber_Security_Threats_Vulnerability_Challenges_and_Proposed_Solution. Truy cập ngày 04/12/2024.

[14] IBM (2023). *Analyzing Files for Embedded Malicious Activity*. IBM Documentation. https://www.ibm.com/docs/en/qsip/7.5?topic=content-analyzing-files-embedded-malicious-activity. Truy cập ngày 04/12/2024.

[15] Canadian Institute for Cybersecurity (n.d.). *CIC-Evasive-PDFMal2022 Dataset*. University of New Brunswick. https://www.unb.ca/cic/datasets/pdfmal-2022.html. Truy cập ngày 10/02/2026.