

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA AN TOÀN THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN  
HỌC PHẦN: AN TOÀN ỨNG DỤNG WEB VÀ CƠ SỞ DỮ LIỆU  
MÃ HỌC PHẦN: INT14105**

**ĐỀ TÀI:**

Học máy trong phân tích log:  
Sử dụng học máy để phân tích log hệ thống nhằm phát hiện tấn công web

**Các sinh viên thực hiện:**

Lý Quốc Khánh – B21DCAT111 (TN)

Phạm Lê Hoàng Anh – B21DCAT

Trần Trọng Mạnh – B21DCAT127

Bùi Duy Thanh – B21DCAT027

**Tên nhóm:** 04

**Tên lớp:** Nhóm 04

**Giảng viên hướng dẫn:** ThS Vũ Minh Mạnh

**HÀ NỘI 2024**

## PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN

TT	Công việc / Nhiệm vụ	SV thực hiện	Thời hạn hoàn thành
1	Trích xuất log, thu nhập dữ liệu - Slide, thuyết trình	Phạm Lê Hoàng Anh	6/10/2024
2	Tiền xử lý: SQL injection, OS command injection. - Báo cáo	Bùi Duy Thanh	21/10/2024
	Tiền xử lý: XXS, Path traversal. - Báo cáo	Trần Trọng Mạnh	25/10/2024
4	Huấn luyện, đánh giá hiệu suất mô hình. - <b>Leader</b>	Lý Quốc Khánh	12/11/2024

## NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ

TT	SV thực hiện	Thái độ tham gia	Mức hoàn thành CV	Kỹ năng giao tiếp	Kỹ năng hợp tác	Kỹ năng lãnh đạo
1	Phạm Lê Hoàng Anh	4	4	4	5	4
2	Bùi Duy Thanh	5	5	4	4	4
3	Trần Trọng Mạnh	4	4	4	5	4
4	Lý Quốc Khánh	5	5	4	4	5

Ghi chú:

- Thái độ tham gia: Đánh giá điểm thái độ tham gia công việc chung của nhóm (từ 0: không tham gia, đến 5: chủ động, tích cực).
- Mức hoàn thành CV: Đánh giá điểm mức độ hoàn thành công việc được giao (từ 0: không hoàn thành, đến 5: hoàn thành xuất sắc).
- Kỹ năng giao tiếp: Đánh giá điểm khả năng tương tác, giao tiếp trong nhóm (từ 0: không hoặc giao tiếp rất yếu, đến 5: giao tiếp xuất sắc).
- Kỹ năng hợp tác: Đánh giá điểm khả năng hợp tác, hỗ trợ lẫn nhau, giải quyết mâu thuẫn, xung đột
- Kỹ năng lãnh đạo: Đánh giá điểm khả năng lãnh đạo (từ 0: không có khả năng lãnh đạo, đến 5: có khả năng lãnh đạo tốt, tổ chức và điều phối công việc trong nhóm hiệu quả).

# MỤC LỤC

DANH MỤC CÁC HÌNH VẼ.....	4
DANH MỤC CÁC BẢNG BIỂU .....	5
DANH MỤC CÁC TỪ VIẾT TẮT.....	6
MỞ ĐẦU .....	7
<b>CHƯƠNG 1. Tổng quan về Ứng dụng Mô hình Học máy trong phát hiện tấn công Web.....</b>	<b>8</b>
1.1 Giới thiệu.....	8
1.1.1 Bối cảnh và Động cơ .....	8
1.1.2 Các loại tấn công web phổ biến .....	8
1.1.3 Mục tiêu.....	16
1.2 Khảo sát nghiên cứu .....	16
1.2.1 Các phương pháp phát hiện tấn công web hiện có.....	16
1.2.2 Ưu điểm của Machine Learning trong phát hiện tấn công web .....	16
1.3 Tổng quan về Rừng ngẫu nhiên (Random Forest) .....	17
1.3.1 Lý Thuyết và Mặt Toán Học .....	17
1.3.2 Ưu điểm và Nhược điểm Của Random Forest .....	19
1.3.3 Ứng Dụng Của Random Forest .....	19
1.4 Kết chương .....	20
<b>CHƯƠNG 2. Thiết kế và Triển khai Mô hình Phát hiện Tấn công .....</b>	<b>21</b>
2.1 Phương pháp nghiên cứu.....	21
2.1.1 Kiến trúc mô hình phát hiện tấn công .....	21
2.1.2 Giai đoạn Huấn luyện.....	21
2.1.3 Giai đoạn Phát hiện .....	26
2.2 Kết chương .....	27
<b>CHƯƠNG 3. Thử nghiệm, Đánh giá và Ứng dụng Thực tiễn .....</b>	<b>28</b>
3.1 Kết quả thực nghiệm .....	28
3.1.1 Hiệu suất mô hình .....	28
3.1.2 Kết quả dự đoán thực nghiệm và hiệu suất của mô hình .....	30
3.1.3 Phân tích hiệu suất mô hình theo từng loại tấn công.....	32
3.2 Đánh giá và Thảo luận .....	36
3.2.1 Ưu điểm và nhược điểm của mô hình .....	36
3.2.2 Khả năng mở rộng và áp dụng thực tiễn .....	37
3.2.3 Giới hạn của nghiên cứu .....	37
3.3 Kết chương .....	38
<b>KẾT LUẬN .....</b>	<b>39</b>

## DANH MỤC CÁC HÌNH VẼ

Hình ảnh 1: Ví dụ về tấn công SQL Injection để trích xuất danh sách tên người dùng và mật khẩu từ cơ sở dữ liệu web .....	9
Hình ảnh 2: Các loại tấn công SQL injection.....	9
Hình ảnh 3: Log web tấn công SQL injection.....	10
Hình ảnh 4: Minh họa mô hình tấn công XSS .....	11
Hình ảnh 5: Đánh giá mức độ ảnh hưởng lỗ hổng XSS .....	11
Hình ảnh 6: Các loại lỗ hổng XSS .....	12
Hình ảnh 7: Log web của tấn công XSS .....	12
Hình ảnh 8: Minh họa tấn công Path Traversal .....	13
Hình ảnh 9: URL bình thường .....	14
Hình ảnh 10: URL của tấn công Path traversal .....	14
Hình ảnh 11: URL của tấn công Path traversal để lấy thông tin file quan trọng .....	14
Hình ảnh 12: Minh họa tấn công OS Command Injection .....	14
Hình ảnh 13: Lỗ hổng CommandInjection trong PHP .....	15
Hình ảnh 14: Log web tấn công OS Command Injection .....	15
Hình ảnh 15: Performane Comparisons of various ML althorithms in IDS .....	17
Hình ảnh 16: Thuật toán Random Forest .....	18
Hình ảnh 17: Mô hình đề xuất cho phát hiện tấn công Web.....	21
Hình ảnh 18: Mô hình chi tiết của phát hiện tấn công Web.....	21
Hình ảnh 19: Đọc dữ liệu vào dataframe d1 .....	22
Hình ảnh 20: Mô tả tổng quan về nguồn dữ liệu.....	22
Hình ảnh 21: Xử lý dữ liệu.....	23
Hình ảnh 22: Format của dữ liệu.....	23
Hình ảnh 23: Trích xuất đặc trưng cho Path traversal.....	24
Hình ảnh 24: Trích xuất đặc trưng cho SQLi.....	24
Hình ảnh 25: Trích xuất đặc trưng cho XSS .....	24
Hình ảnh 26: Trích xuất đặc trưng cho OS injection .....	24
Hình ảnh 27: Đọc dữ liệu vào dataframe .....	25
Hình ảnh 28: Chia tập dữ liệu để huấn luyện .....	25
Hình ảnh 29: Xây dựng tham số cho Random Forest .....	26
Hình ảnh 30: Lưu mô hình đã huấn luyện.....	26
Hình ảnh 31: Công thức tính Accuracy.....	28
Hình ảnh 32: Công thức tính Precision .....	29
Hình ảnh 33: Công thức tính Recall.....	29
Hình ảnh 34: Công thức tính F1-score .....	29
Hình ảnh 35: Kết quả dự đoán.....	30
Hình ảnh 36: Kết quả dự đoán ở dạng file sheet .....	30
Hình ảnh 37: Tỷ lệ giữa các hành vi bình thường và các cuộc tấn công.....	31
Hình ảnh 38: Sử dụng mô hình và đánh giá kết quả .....	32

Hình ảnh 39: Hiệu suất mô hình đối với SQL injection.....	33
Hình ảnh 40: Hiệu suất mô hình đối với XSS.....	34
Hình ảnh 41: Hiệu suất mô hình đối với Path Traversal .....	35
Hình ảnh 42: Hiệu suất mô hình đối với OS Command Injection .....	36

## DANH MỤC CÁC BẢNG BIỂU

Bảng 1: Đánh giá trên từng loại tấn công.....	32
Bảng 2: Chỉ số Precision, Recall, F1-score của Path Traversal .....	35

## DANH MỤC CÁC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Thuật ngữ tiếng Anh/Giải thích</b>	<b>Thuật ngữ tiếng Việt giải thích</b>
API	Application Programming Interface	Giao diện lập trình ứng dụng (API)
CART	Classification and Regression Trees	Cây phân loại và hồi quy
DNS	Domain Name System	Hệ thống tên miền
FP	False Positive	Dự đoán sai là tấn công (False Positive)
FN	False Negative	Dự đoán sai là bình thường (False Negative)
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản (Dùng để xây dựng các trang web)
IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
IPS	Intrusion Prevention System	Hệ thống phòng chống xâm nhập
IP	Internet Protocol	Giao thức Internet
LDA	Linear Discriminant Analysis	Phân tích phân biệt tuyến tính
MSE	Mean Squared Error	Lỗi bình phương trung bình
OS	Operating System	Hệ điều hành
PT	Path Traversal	Tấn công vượt qua đường dẫn, là kỹ thuật tấn công khai thác các đường dẫn hệ thống tệp
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
SQLi	SQL Injection	Tấn công SQL Injection, là kỹ thuật tấn công vào ứng dụng web thông qua các câu lệnh SQL độc hại
SVM	Support Vector Machine	Máy vector hỗ trợ, một thuật toán học máy thường được sử dụng cho phân loại dữ liệu
TN	True Negative	Dự đoán đúng là bình thường (True Negative)
TP	True Positive	Dự đoán đúng là tấn công (True Positive)
URI	Uniform Resource Identifier	Định danh tài nguyên đồng nhất, là một chuỗi các ký tự dùng để xác định tài nguyên trên Internet
XSS	Cross-Site Scripting	Tấn công chèn mã JavaScript độc hại vào ứng dụng web

## MỞ ĐẦU

Trong kỷ nguyên số hóa, các hệ thống web đóng vai trò quan trọng trong việc cung cấp dịch vụ, lưu trữ và trao đổi thông tin. Tuy nhiên, đi kèm với sự phát triển nhanh chóng của công nghệ là sự gia tăng không ngừng của các mối đe dọa an ninh mạng. Các cuộc tấn công vào hệ thống web, bao gồm tấn công từ chối dịch vụ (DDoS), khai thác lỗ hổng, tấn công SQL Injection, và nhiều loại hình tấn công khác, đã gây ra những hậu quả nghiêm trọng về tài chính và uy tín đối với các tổ chức và cá nhân.

Một trong những thách thức lớn nhất trong lĩnh vực an ninh mạng là khả năng phát hiện sớm và ngăn chặn các cuộc tấn công ngay từ khi chúng bắt đầu. Các phương pháp phát hiện truyền thống dựa trên luật hoặc chữ ký (signature-based) thường không đủ hiệu quả để đối phó với các loại tấn công ngày càng tinh vi, đặc biệt là các hình thức tấn công mới chưa từng xuất hiện trước đó.

Trước tình hình đó, học máy (Machine Learning) đã trở thành một hướng tiếp cận đầy tiềm năng trong việc phát hiện tấn công web. Với khả năng phân tích và học hỏi từ các tập dữ liệu lớn, học máy có thể phát hiện ra các hành vi bất thường tiềm ẩn và thậm chí dự đoán các mối đe dọa chưa biết trước. Đặc biệt, việc áp dụng học máy vào phân tích log hệ thống – nguồn dữ liệu ghi lại toàn bộ hoạt động của hệ thống – đã mở ra một hướng đi mới, giúp nâng cao hiệu quả phát hiện và phản ứng nhanh với các cuộc tấn công.

Mục tiêu của nghiên cứu này là:

1. Khảo sát các loại tấn công web phổ biến và các dấu hiệu tiềm ẩn trong log hệ thống.
2. Phát triển mô hình học máy nhằm phân tích và phát hiện các hành vi tấn công dựa trên dữ liệu log.
3. Đánh giá hiệu suất và tính thực tiễn của mô hình trong việc phát hiện tấn công web.

Với sự kết hợp giữa các thuật toán học máy tiên tiến và các phương pháp phân tích log, nghiên cứu này kỳ vọng sẽ đóng góp vào việc nâng cao khả năng bảo vệ các hệ thống web, đáp ứng yêu cầu thực tế trong lĩnh vực an ninh mạng ngày nay.

# CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG MÔ HÌNH HỌC MÁY TRONG PHÁT HIỆN TẤN CÔNG WEB

## 1.1 Giới thiệu

### 1.1.1 Bối cảnh và Động cơ

Trong thời đại số, khi các ứng dụng web ngày càng phát triển và đóng vai trò quan trọng trong hầu hết mọi lĩnh vực, từ kinh doanh đến quản lý thông tin, vấn đề bảo mật web trở nên ngày càng cấp thiết. Các cuộc tấn công mạng nhằm vào các ứng dụng web ngày càng gia tăng và tinh vi, gây ra hậu quả nghiêm trọng bao gồm rò rỉ dữ liệu, phá hủy hệ thống, và ảnh hưởng đến uy tín của tổ chức. Một số loại tấn công phổ biến nhất bao gồm: *SQLi*, *XSS*, *Path Traversal*, *OS Command Injection*.

Những cuộc tấn công này có thể giúp kẻ tấn công vượt qua cơ chế xác thực của hệ thống web, thực hiện các thay đổi không hợp lệ đối với nội dung và cơ sở dữ liệu web, lấy dữ liệu quan trọng từ cơ sở dữ liệu ứng dụng web, đánh cắp thông tin nhạy cảm từ máy chủ web và người dùng, thậm chí kiểm soát hoàn toàn các máy chủ web và/hoặc máy chủ cơ sở dữ liệu.

Phát hiện và ngăn chặn các cuộc tấn công này kịp thời là một thách thức lớn đối với các tổ chức, đặc biệt khi các cuộc tấn công liên tục phát triển và có thể dễ dàng vượt qua các cơ chế bảo mật truyền thống.

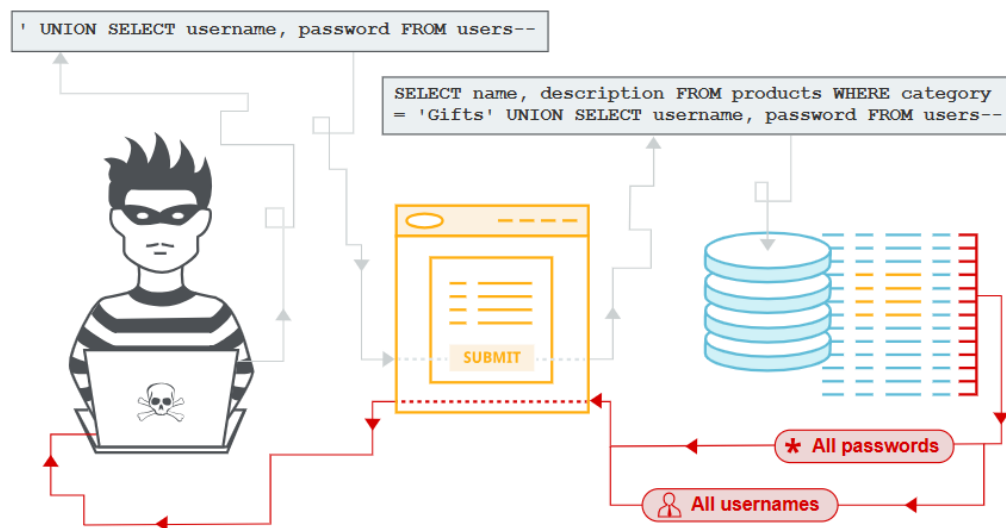
Trong bối cảnh này, **phân tích log web** nổi lên như một trong những phương pháp hiệu quả để phát hiện các hành vi đáng ngờ và ngăn chặn các cuộc tấn công. Log web ghi lại tất cả các truy cập và yêu cầu đến ứng dụng, giúp cho việc theo dõi và phát hiện các mẫu hành vi bất thường, từ đó có thể xác định các hoạt động tấn công tiềm ẩn. Việc phân tích log web không chỉ hỗ trợ phát hiện tấn công mà còn cung cấp thông tin chi tiết về cách thức hoạt động của các cuộc tấn công, giúp các tổ chức cải thiện hệ thống phòng thủ của mình.

### 1.1.2 Các loại tấn công web phổ biến

#### 1.1.2.1 SQL Injection

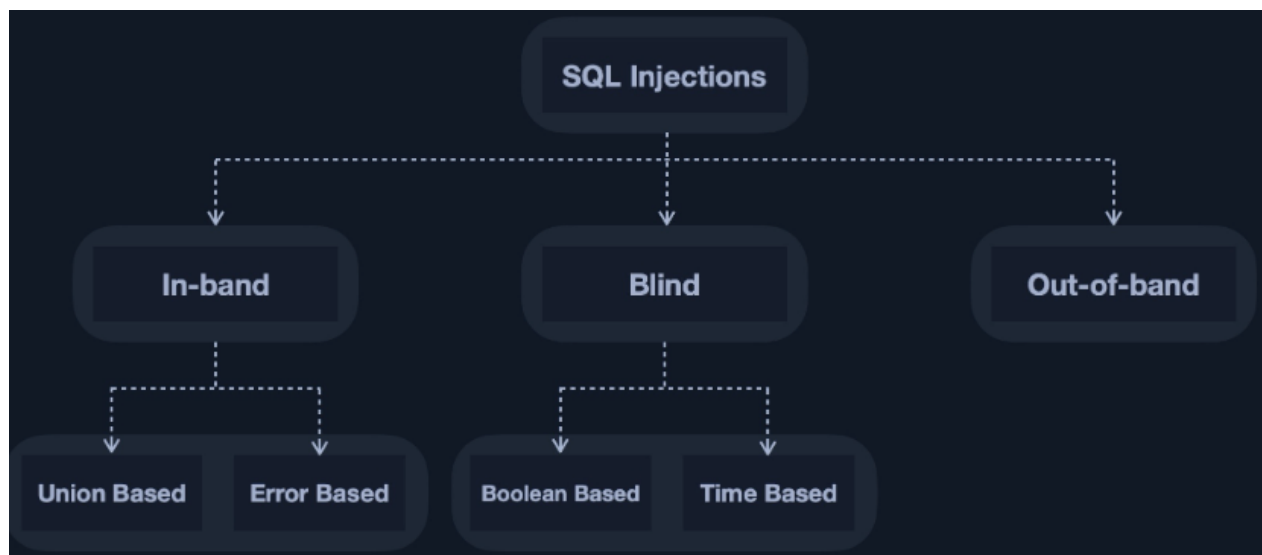
Đây là một trong những hình thức tấn công phổ biến nhất, cho phép kẻ tấn công chèn mã SQL độc hại vào các truy vấn cơ sở dữ liệu. Khi thành công, SQL Injection có thể cho phép kẻ tấn công truy cập vào thông tin nhạy cảm hoặc thậm chí thay đổi dữ liệu trong cơ sở dữ liệu.





Hình ảnh 1: Ví dụ về tấn công SQL Injection để trích xuất danh sách tên người dùng và mật khẩu từ cơ sở dữ liệu web

## - Các kiểu tấn công SQLi



Hình ảnh 2: Các loại tấn công SQL injection

Trong những trường hợp đơn giản, kết quả của cả truy vấn dự định thực thi hợp lệ và truy vấn bị thay đổi có thể được hiển thị trực tiếp trên giao diện người dùng, và chúng ta có thể đọc chúng trực tiếp. Điều này được gọi là **In-band SQL Injection**, và nó bao gồm hai loại:

- **Union Based SQL Injection:** Chúng ta có thể phải chỉ định vị trí cụ thể (ví dụ: cột) để kết quả được hiển thị. Truy vấn sẽ điều hướng kết quả đến vị trí đó để được hiển thị.
- **Error Based SQL Injection:** Được sử dụng khi các lỗi PHP hoặc SQL được hiển thị trên giao diện người dùng. Trong trường hợp này, chúng ta có thể cố ý gây ra một lỗi SQL để trả về kết quả của truy vấn.

Trong những trường hợp phức tạp hơn, chúng ta có thể không thấy kết quả được hiển thị trực tiếp, nên phải sử dụng logic SQL để truy xuất kết quả từng ký tự một. Điều này được gọi là **Blind SQL Injection** (SQL Injection mù), và nó cũng bao gồm hai loại:

- Boolean Based SQL Injection: Chúng ta sử dụng các câu lệnh điều kiện SQL để kiểm soát xem trang có trả về bất kỳ kết quả nào hay không (tức là phản hồi truy vấn gốc) khi câu lệnh điều kiện trả về giá trị true.
- Time Based SQL Injection: Sử dụng các câu lệnh điều kiện SQL để trì hoãn phản hồi của trang nếu câu lệnh điều kiện trả về giá trị true, bằng cách sử dụng hàm Sleep().

Cuối cùng, trong một số trường hợp, chúng ta có thể không có quyền truy cập trực tiếp vào kết quả, nên cần điều hướng kết quả đến một vị trí từ xa (ví dụ: bản ghi DNS) và sau đó cố gắng truy xuất kết quả từ đó. Điều này được gọi là **Out-of-band SQL Injection**.

#### - 1 ví dụ Log web của tấn công SQL Injection

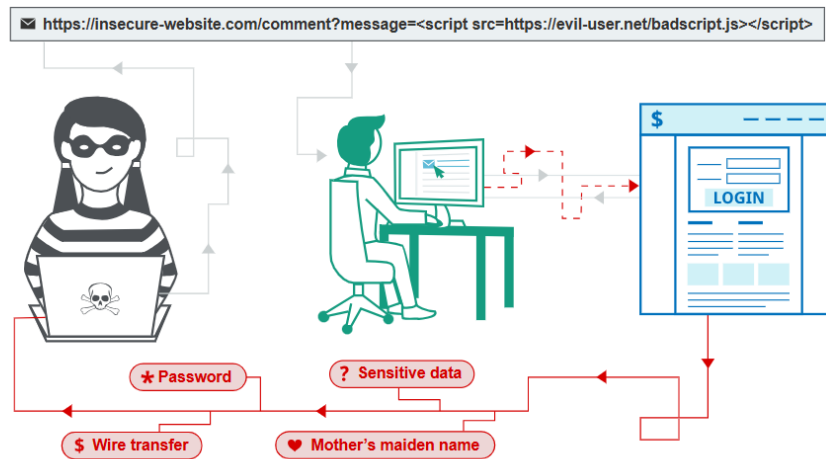
```
<?xml version="1.0" encoding="utf-8"?><dataset author="CSIC" name="Tienda2012">
  <sample id="30000">
    <request>
      <method>POST</method>
      <protocol>HTTP/1.1</protocol>
      <path><![CDATA[/tienda1/publico/autenticar.jsp]]></path>
      <headers><![CDATA[Accept-Encoding: identity
Content-Length: 68
Accept-Language: en-us,en;q=0.5
Connection: close
User-Agent: sqlmap/1.0-dev (r4198) (http://sqlmap.sourceforge.net)"]>
      </headers>
      <body><![CDATA[login=61%27%20OR%20%2761%27=%2761&pwd=FrAmE30.&remember=&modo=entrar]]></body>
    </request>
    <label>
      <type>attack</type>
      <attack>SQLi</attack>
    </label>
```

Hình ảnh 3: Log web tấn công SQL injection

Sau đó qua quá trình tiền xử lý để thu được 1 URL nhằm đại diện cho cuộc tấn công đó. Nội dung này sẽ được đề cập ở Chương 2.

#### 1.1.2.2 Cross-Site Scripting (XSS)

Một ứng dụng web điển hình hoạt động bằng cách nhận mã HTML từ máy chủ backend và hiển thị nó trên trình duyệt của người dùng. Khi một ứng dụng web không kiểm tra và lọc đúng cách đầu vào của người dùng, kẻ tấn công có thể chèn mã JavaScript độc hại vào một trường đầu vào (ví dụ: bình luận hoặc phản hồi). Khi người dùng khác xem cùng một trang, họ sẽ vô tình thực thi mã JavaScript độc hại đó.



Hình ảnh 4: Minh họa mô hình tấn công XSS

Lỗ hổng XSS chỉ được thực thi trên phía trình duyệt (client-side) và không ảnh hưởng trực tiếp đến máy chủ backend. Chúng chỉ ảnh hưởng đến người dùng thực thi lỗ hổng đó. Mặc dù tác động trực tiếp của lỗ hổng XSS đến máy chủ backend có thể tương đối thấp, nhưng chúng rất phổ biến trong các ứng dụng web. Vì vậy, rủi ro từ XSS được đánh giá ở mức trung bình (tác động thấp + xác suất cao = rủi ro trung bình), và chúng ta nên luôn cố gắng giảm thiểu rủi ro này bằng cách phát hiện, khắc phục và ngăn chặn chủ động các loại lỗ hổng này.



Hình ảnh 5: Đánh giá mức độ ảnh hưởng lỗ hổng XSS

- Có 3 loại lỗ hổng XSS chính: **Stored XSS**, **Reflected XSS** và **DOM-based XSS**

Kiểu	Sự miêu tả
<b>Stored (Persistent) XSS</b>	Loại XSS quan trọng nhất, xảy ra khi đầu vào của người dùng được lưu trữ trên cơ sở dữ liệu phụ trợ và sau đó được hiển thị khi truy xuất (ví dụ: bài đăng hoặc nhận xét)
<b>Reflected (Non-Persistent) XSS</b>	Xảy ra khi đầu vào của người dùng được hiển thị trên trang sau khi được xử lý bởi máy chủ phụ trợ, nhưng không được lưu trữ (ví dụ: kết quả tìm kiếm hoặc thông báo lỗi)
<b>DOM-based XSS</b>	Một loại XSS không liên tục khác xảy ra khi đầu vào của người dùng được hiển thị trực tiếp trong trình duyệt và được xử lý hoàn toàn ở phía máy khách, mà không đến được máy chủ phụ trợ (ví dụ: thông qua các tham số HTTP phía máy khách hoặc thẻ neo)

Hình ảnh 6: Các loại lỗ hổng XSS

Lỗ hổng XSS có thể hỗ trợ một loạt các cuộc tấn công, bao gồm bất kỳ điều gì có thể được thực thi qua mã JavaScript trình duyệt. Một ví dụ cơ bản là khiến người dùng mục tiêu vô tình gửi cookie phiên của họ đến máy chủ của kẻ tấn công. Một ví dụ khác là khiến trình duyệt của người dùng thực hiện các lệnh API dẫn đến hành động độc hại, chẳng hạn như thay đổi mật khẩu của người dùng thành mật khẩu do kẻ tấn công chọn.

Vì các cuộc tấn công XSS thực thi mã JavaScript trong trình duyệt, chúng bị giới hạn trong môi trường JavaScript của trình duyệt (ví dụ: **V8** trong Chrome). Chúng không thể thực thi mã trên toàn hệ thống, như thực thi mã cấp hệ thống. Trên các trình duyệt hiện đại, XSS cũng bị giới hạn trong cùng một miền (domain) của trang web để bị tổn thương. Tuy nhiên, việc có thể thực thi JavaScript trong trình duyệt của người dùng vẫn có thể dẫn đến một loạt các cuộc tấn công khác nhau như đã đề cập ở trên.

- Log web được thể hiện ở dạng file XML của tấn công XSS

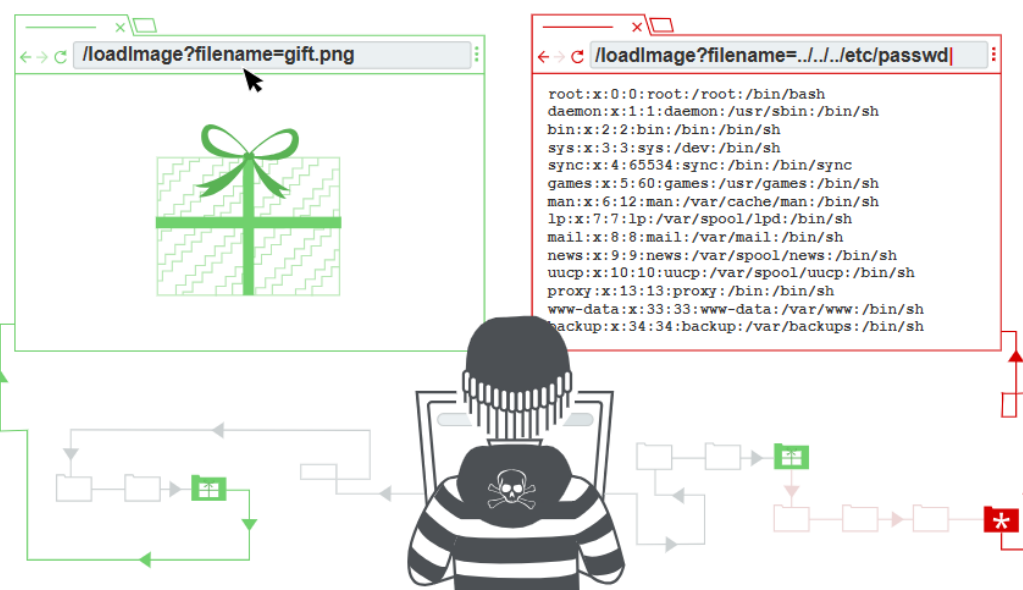
```
<protocol>HTTP/1.1</protocol>
<path><![CDATA[/tienda1/miembros/editar.jsp]]></path>
<headers><![CDATA[User-Agent: Googlebot/2.1 (+http://www.google.com/bot.html)
Host: localhost:8080
Cookie: <script>alert('207a67fe67d047dd03b5fd370e0db66e')</script>
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg
Connection: Keep-Alive
Content-type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 4074
Expect: 100-continue
]]></headers>
<body><![CDATA[modo=insertar&login=c2&password=c2&nombre=carlos&apellidos=perez&email=perez@yahoo.com&dni=23
4534572&provincia=1&cp=12354&B1=Confirmar&ciudad=Madrid&ntc=1234567890123456&direccion=%35%31%2c%35%32%2c%33%3
7%2c%35%30%2c%35%33%2c%35%30%2c%36%37%2c%35%34%2c%35%30%2c%33%37%2c%35%30%2c%35%33%2c%35%30%2c%36%37%2c%35%34%
%3c%34%38%2c%33%37%2c%35%30%2c%35%33%2c%35%30%2c%36%37%2c%34%39%2c%34%38%2c%35%33%2c%33%37%2c%35%30%2c%35%33%2c%35%30%2c%36%37%2c%34%
</request>
<label>
  <type>attack</type>
  <attack>XSS</attack>
</label>
</sample>
<sample id="42021">
  <request>
    <method>POST</method>
    <protocol>HTTP/1.1</protocol>
    <path><![CDATA[/tienda1/miembros/editar.jsp]]></path>
    <headers><![CDATA[User-Agent: Googlebot/2.1 (+http://www.google.com/bot.html)
Host: localhost:8080
Cookie: <script>alert('ea06c7def5190ac43cfb8229d5e2c5ec')</script>
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg
Connection: Keep-Alive
```

Hình ảnh 7: Log web của tấn công XSS

### 1.1.2.3 Path Traversal

Một cuộc tấn công Path Traversal (hay còn gọi là Directory Traversal) nhằm mục đích truy cập các tệp và thư mục được lưu trữ bên ngoài thư mục gốc của web. Bằng cách thao tác các biến tham chiếu đến tệp với các chuỗi “dot-dot-slash (../)” và các biến thể của nó, hoặc bằng cách sử dụng các đường dẫn tệp tuyệt đối, kẻ tấn công có thể truy cập vào các tệp và thư mục tùy ý trên hệ thống tệp, bao gồm mã nguồn ứng dụng, tệp cấu hình, và các tệp hệ thống quan trọng. Cần lưu ý rằng việc truy cập các tệp bị giới hạn bởi quyền kiểm soát truy cập của hệ thống (chẳng hạn như trường hợp các tệp bị khóa hoặc đang được sử dụng trên hệ điều hành Microsoft Windows).

Cuộc tấn công này còn được gọi là “dot-dot-slash”, “directory traversal”, “directory climbing” và “backtracking”.



Hình ảnh 8: Minh họa tấn công Path Traversal

### - Cách xác định lỗ hổng Path Traversal

- Hiểu rõ cách hệ điều hành xử lý các tên tệp mà nó nhận được.
- Không lưu trữ các tệp cấu hình nhạy cảm trong thư mục gốc web.
- Đối với các máy chủ Windows IIS, thư mục gốc web không nên đặt trên ổ hệ thống, để ngăn chặn truy hồi thư mục ngược về các thư mục hệ thống.

### - Các biến thể tấn công

- + Mã hóa và mã hóa kép các chuỗi “dot-dot-slash (../)” và các biến thể của nó nhằm vượt qua các bộ lọc.
- + Mã hóa phần trăm (URL encoding): Các trình xử lý web thường thực hiện một cấp độ giải mã các giá trị được mã hóa từ form và URL.
- + Tùy chỉnh kí tự “/” và “\” đối với các hệ điều hành cụ thể
- + Chèn null byte(%00) để bypass phần hậu tố URL

### - Ví dụ

```
http://some_site.com.br/get-files.jsp?file=report.pdf
http://some_site.com.br/get-page.php?home=aaa.html
http://some_site.com.br/some-page.asp?page=index.html
```

Hình ảnh 9: URL bình thường

```
http://some_site.com.br/get-files?file=../../../../some dir/some file
http://some_site.com.br/../../../../some dir/some file
```

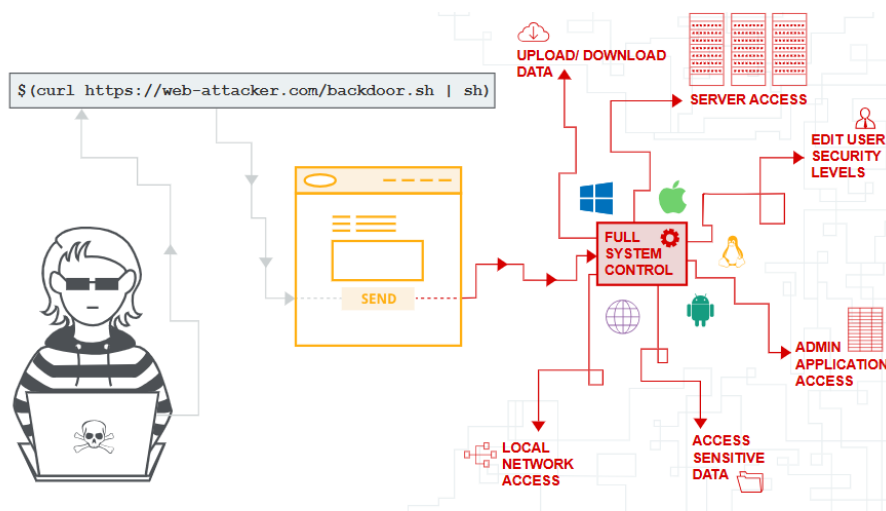
Hình ảnh 10: URL của tấn công Path traversal

```
http://some_site.com.br/../../../../etc/shadow
http://some_site.com.br/get-files?file=/etc/passwd
```

Hình ảnh 11: URL của tấn công Path traversal để lấy thông tin file quan trọng

#### 1.1.2.4 OS Command Injection:

Các lỗ hổng Injection được xếp thứ 3 trong *OWASP Top 10 Critical Web Application Security Risks*, bởi vì mức độ tác động nghiêm trọng và mức độ phổ biến của chúng. Injection xảy ra khi dữ liệu đầu vào do người dùng kiểm soát bị hiểu nhầm là một phần của truy vấn web hoặc mã thực thi, dẫn đến việc thay đổi kết quả truy vấn theo hướng có lợi cho kẻ tấn công.



Hình ảnh 12: Minh họa tấn công OS Command Injection

Đối với **OS Command Injection**, dữ liệu đầu vào của người dùng phải trực tiếp hoặc gián tiếp được đưa vào (hoặc tác động) một truy vấn web thực thi các lệnh hệ thống. Các ngôn ngữ lập trình web đều có các hàm cho phép nhà phát triển thực thi lệnh hệ điều hành

trực tiếp trên máy chủ backend. Điều này thường được sử dụng cho nhiều mục đích, như cài đặt plugin hoặc thực thi ứng dụng cụ thể.

Ví dụ, một ứng dụng web viết bằng PHP có thể sử dụng các hàm như `exec`, `system`, `shell_exec`, `passthru`, hoặc `popen` để thực thi lệnh trên máy chủ backend, mỗi hàm có một trường hợp sử dụng khác nhau. Mã sau đây là ví dụ về lỗ hổng **command injection** trong PHP:

```
<?php
if (isset($_GET['filename'])) {
    system("touch /tmp/" . $_GET['filename'] . ".pdf");
}
?>
```

Hình ảnh 13: Lỗ hổng Command Injection trong PHP

Giả sử ứng dụng web này có tính năng cho phép người dùng tạo một tài liệu .pdf mới trong thư mục /tmp với tên file do người dùng cung cấp và sau đó được sử dụng để xử lý tài liệu. Tuy nhiên, vì tham số filename từ yêu cầu GET được sử dụng trực tiếp trong lệnh touch (mà không được kiểm tra hoặc lọc ký tự), ứng dụng web sẽ trở nên dễ bị tấn công **Command Injection**. Lỗ hổng này có thể bị khai thác để thực thi các lệnh hệ thống tùy ý trên máy chủ backend.

Ngoài ra, các ngôn ngữ lập trình khác cũng có những hàm tương tự và, nếu tồn tại lỗ hổng, có thể bị khai thác theo cách tương tự. Hơn nữa, lỗ hổng **Command Injection** không chỉ giới hạn ở các ứng dụng web mà còn có thể ảnh hưởng đến các chương trình nhị phân khác và các ứng dụng client nặng nếu chúng truyền dữ liệu đầu vào không kiểm tra tới hàm thực thi lệnh hệ thống, điều này cũng có thể bị khai thác bằng các phương pháp tương tự.

```
<protocol>HTTP/1.1</protocol>
<path><![CDATA[/tienda1/miembros/editar.jsp]]></path>
<headers><![CDATA[User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.18) Gecko/20110615 Ubuntu/10.04 (lucid) Firefox/3.6.18 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-us,en;q=0.5 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7 Keep-Alive: 115 Proxy-Connection: keep-alive Referer: http://localhost:8080/tienda1/miembros/editar.jsp Cookie: JSESSIONID=99BCC669E26661C74042EB0EC9F2FEF7; usuario=815 Content-Type: application/x-www-form-urlencoded Content-length: 238 Host: localhost:8080 ]]></headers>
<body><![CDATA[modo=insertar&login=pepe&password=pepe&nombre=%3C%21--%23EXEC&cmd%30%22ls+%2f%22--%3E&apellidos=perez&email=pepe@pepe.com]]></body>
</request>
<label>
<type>attack</type>
<attack>$I</attack>
</label>
</sample>
<sample id="47993">
<request>
<method>GET</method>
<protocol>HTTP/1.1</protocol>
<path><![CDATA[/tienda1/publico/vaciar.jsp]]></path>
<query><![CDATA[b2=%3C!--%23EXEC%20cmd=%22dir%20%5C%22--%3E]]></query>
<headers><![CDATA[User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)]></headers>
Pragma: no-cache
Content-Type: application/x-www-form-urlencoded
Content-length: 0
```

Hình ảnh 14: Log web tấn công OS Command Injection



### 1.1.3 Mục tiêu

Mục tiêu của báo cáo này là phát triển một mô hình phát hiện tấn công web “**phổ biến**” dựa trên machine learning, sử dụng các bản ghi log web làm dữ liệu huấn luyện.

## 1.2 Khảo sát nghiên cứu

### 1.2.1 Các phương pháp phát hiện tấn công web hiện có

Phát hiện tấn công web là một lĩnh vực có nhiều nghiên cứu và phát triển trong suốt nhiều năm qua, với một số phương pháp truyền thống và hiện đại được áp dụng để ngăn chặn các loại tấn công. Các phương pháp này có thể được chia thành các nhóm chính như sau:

- *Các đề xuất trong nhóm dựa trên lọc dữ liệu đầu vào* sử dụng tập hợp các quy tắc, chữ ký, hoặc kỹ thuật để lọc và xác thực dữ liệu đầu vào nhằm phát hiện và ngăn chặn các cuộc tấn công web. Một số đề xuất tiêu biểu trong nhóm này là *OWAP Core Rules Set* được phát triển bởi dự án OWASP để phát hiện nhiều loại tấn công web trong danh sách OWASP top 10 với tỷ lệ báo động sai thấp. Các đề xuất trong nhóm lọc dữ liệu đầu vào có thể phát hiện tấn công web một cách hiệu quả. Tuy nhiên, chúng yêu cầu xây dựng thủ công và cập nhật thường xuyên các quy tắc hoặc mô hình phát hiện.
- *Các phương pháp phát hiện tấn công web trong nhóm dựa trên bất thường*: Trước tiên cần xây dựng một "hồ sơ" của ứng dụng web trong điều kiện làm việc bình thường và sau đó theo dõi các hoạt động của các ứng dụng web. Nếu có bất kỳ sự khác biệt đáng kể nào giữa các hoạt động hiện tại và những gì được lưu trữ trong "hồ sơ", một cảnh báo tấn công sẽ được đưa ra. Phương pháp đề xuất được báo cáo có tỷ lệ phát hiện cao cũng như tỷ lệ báo động giả thấp.
- *Nhóm dựa trên học máy sử dụng các thuật toán học máy* để xây dựng các mô hình phát hiện và sau đó sử dụng những mô hình này để phát hiện các cuộc tấn công web có thể xảy ra. Các thuật toán học máy được sử dụng có thể là các phương pháp học truyền thống, chẳng hạn như Naive Bayes, Decision Tree, SVM, Random Forest, hoặc các phương pháp Deep Learning như CNN và RNN. Tuy nhiên, Deep Learning thường tốn kém và có thể không phù hợp cho phát hiện tấn công web thời gian thực.

### 1.2.2 Ưu điểm của Machine Learning trong phát hiện tấn công web

Việc áp dụng Machine Learning vào phát hiện tấn công web mang lại một số lợi ích quan trọng như:

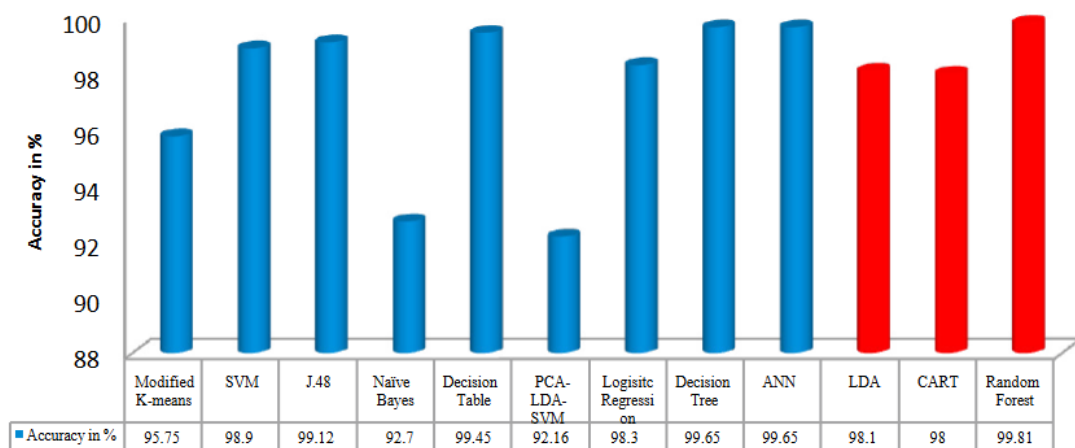
- *Tự động hóa*: Machine learning giúp tự động hóa quy trình phát hiện tấn công mà không cần can thiệp liên tục của con người. Hệ thống có thể tự động học từ dữ liệu và cải thiện khả năng phân loại các yêu cầu web mà không yêu cầu phải tạo ra các quy tắc thủ công.
- *Phát hiện các mẫu tấn công mới*: Một trong những lợi ích chính của machine learning là khả năng phát hiện các mẫu hành vi bất thường hoặc chưa được biết đến trước đây. Điều này giúp hệ thống có khả năng phát hiện cả các tấn công zero-day và các biến thể của tấn công đã biết.



- *Khả năng tùy chỉnh cao:* Các mô hình machine learning có thể dễ dàng được tùy chỉnh và huấn luyện lại để phù hợp với các loại tấn công cụ thể hoặc môi trường ứng dụng khác nhau, giúp gia tăng hiệu quả phát hiện tấn công trong các bối cảnh khác nhau.
- *Giảm thiểu cảnh báo giả (False Positives):* Với dữ liệu huấn luyện đủ lớn và phong phú, machine learning có thể cải thiện độ chính xác, giảm thiểu các cảnh báo sai, giúp các chuyên gia bảo mật tập trung vào các mối đe dọa thực sự.

### 1.3 Tổng quan về Rừng ngẫu nhiên (Random Forest)

Theo một nghiên cứu về việc áp dụng các thuật toán học máy trong hệ thống phát hiện xâm nhập (IDS), kết quả thử nghiệm cho thấy thuật toán **Random Forest (RF)** đạt được độ chính xác cao nhất (99,65%) so với LDA (98,1%) và CART (98%). Điều này chứng tỏ RF là lựa chọn tốt hơn khi cần phân loại và nhận dạng các mối đe dọa an ninh mạng. Trong khi đó, các thuật toán khác như LDA và CART cũng có hiệu suất khá tốt, nhưng RF vẫn vượt trội trong việc đạt được kết quả chính xác hơn. (T. Saranya et al., 2020)



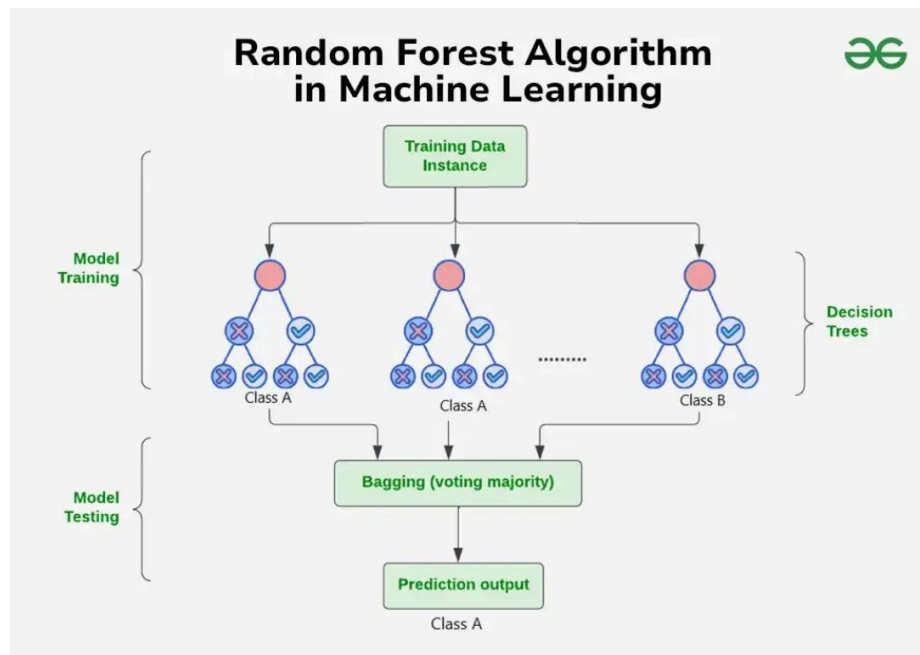
Hình ảnh 15: Performane Comparisons of various ML althorithms in IDS

Các kết quả phân loại từ bộ dữ liệu KDD Cup sử dụng các thuật toán LDA, CART và RF được minh họa trong các hình ảnh trong bài báo gốc (xem Hình ảnh 5). Các so sánh về độ chính xác giữa các thuật toán này cũng cho thấy **RF** mang lại kết quả chính xác hơn, điều này làm cho RF trở thành một sự lựa chọn đáng tin cậy cho việc phân loại các cuộc tấn công (T. Saranya et al., 2020)

#### 1.3.1 Lý Thuyết và Mặt Toán Học

##### 1.3.1.1 Cấu Trúc và Hoạt Động Của Random Forest

Rừng ngẫu nhiên (RF) là một thành viên trong chuỗi các thuật toán cây quyết định (Decision Tree). Ý tưởng của rừng ngẫu nhiên là tạo ra một số cây quyết định. Những cây quyết định này sẽ chạy và sản xuất kết quả độc lập. Câu trả lời được dự đoán bởi số lượng lớn nhất của các cây quyết định sẽ được chọn bởi rừng ngẫu nhiên. Để đảm bảo rằng các cây quyết định không giống nhau, rừng ngẫu nhiên chọn ngẫu nhiên một tập hợp các đặc điểm cho mỗi nút. Các tham số còn lại được sử dụng trong rừng ngẫu nhiên giống như trong các cây quyết định.



Hình ảnh 16: Thuật toán Random Forest

#### 1.3.1.2 Bootstrap Aggregating (Bagging)

Để tạo ra các tập con dữ liệu khác nhau, Random Forest sử dụng một kỹ thuật gọi là Bootstrap Aggregating, hay còn gọi là bagging. Với mỗi cây trong rừng, một tập con dữ liệu được chọn ngẫu nhiên từ tập dữ liệu huấn luyện ban đầu bằng cách lấy mẫu có hoàn lại. Kỹ thuật này giúp mỗi cây quyết định nhận một tập dữ liệu hơi khác nhau, giảm khả năng cùng mắc lỗi giống nhau và giúp mô hình đạt được độ ổn định cao hơn.

#### 1.3.1.3 Chọn Đặc Trưng Ngẫu Nhiên

Trong mỗi cây quyết định của Random Forest, khi thực hiện chia nhánh tại mỗi nút, thay vì xem xét tất cả các đặc trưng, Random Forest chỉ chọn một tập con ngẫu nhiên các đặc trưng để chia. Điều này giúp các cây trở nên khác nhau hơn, giảm mối tương quan giữa các cây, từ đó làm tăng độ chính xác tổng thể của rừng cây và giảm hiện tượng overfitting.

#### 1.3.1.4 Thuật Toán và Quá Trình Huấn Luyện

- Bước 1: Tạo tập con dữ liệu bằng phương pháp lấy mẫu có hoàn lại từ tập dữ liệu huấn luyện gốc.
- Bước 2: Huấn luyện một cây quyết định cho mỗi tập con dữ liệu với một tập con các đặc trưng ngẫu nhiên.
- Bước 3: Lặp lại Bước 1 và Bước 2 để tạo ra một số lượng lớn cây quyết định (thường từ hàng chục đến hàng trăm cây).
- Bước 4: Kết hợp kết quả của tất cả các cây để đưa ra dự đoán cuối cùng.

#### 1.3.1.5 Toán Học Cơ Bản Trong Random Forest

Giả sử một tập dữ liệu có  $N$  quan sát và  $M$  đặc trưng. Với mỗi cây:

- Một tập dữ liệu có kích thước  $N$  được lấy mẫu từ tập dữ liệu ban đầu bằng phương pháp lấy mẫu có hoàn lại.

- Với mỗi nút chia, Random Forest chỉ xét  $m$  đặc trưng ngẫu nhiên (với  $m < M$ ) để tìm đặc trưng tối ưu cho việc chia nhánh.

#### 1.3.1.6 Các Tham Số Quan Trọng

Một số tham số cần lưu ý khi sử dụng Random Forest:

- **n\_estimators:** Số lượng cây trong rừng, càng nhiều cây thì mô hình càng ổn định, nhưng thời gian tính toán cũng sẽ tăng lên.
- **max\_features:** Số lượng đặc trưng tối đa để xét khi thực hiện chia nhánh tại mỗi nút.
- **max\_depth:** Độ sâu tối đa của cây quyết định, giúp ngăn mô hình học quá sâu và dẫn đến overfitting.

#### 1.3.2 Ưu điểm và Nhược điểm Của Random Forest

##### \*\* Ưu điểm

- *Khả năng tổng quát hóa tốt:* Vì Random Forest kết hợp nhiều cây, nên mô hình có khả năng tổng quát tốt hơn và ít bị overfitting so với cây quyết định đơn lẻ.
- *Khả năng xử lý dữ liệu lớn và phức tạp:* Mô hình có thể xử lý tốt các tập dữ liệu lớn, đa dạng và có tính phi tuyến cao.
- *Tính ổn định cao:* Nhờ quá trình bagging và lựa chọn đặc trưng ngẫu nhiên, Random Forest không nhạy cảm với nhiễu và các thay đổi nhỏ trong dữ liệu.

##### \*\* Nhược điểm

- *Tính giải thích hạn chế:* So với một cây quyết định đơn lẻ, Random Forest khó giải thích và truy vết từng quyết định cụ thể hơn.
- *Yêu cầu tài nguyên:* Random Forest cần nhiều tài nguyên tính toán hơn so với một cây quyết định đơn lẻ, đặc biệt khi số lượng cây lớn.

#### 1.3.3 Ứng Dụng Của Random Forest

##### 1.3.3.1 Phát Hiện Tấn Công Web

Trong lĩnh vực an toàn thông tin, Random Forest được áp dụng để phát hiện các cuộc tấn công thông qua phân tích log web. Với khả năng học từ dữ liệu không đồng nhất và tính ổn định, Random Forest có thể phân loại các yêu cầu bất thường, chẳng hạn như XSS, SQL injection, Path Traversal và OS Command Injection trong các log web.

##### 1.3.3.2 Phát Hiện Gian Lận

Random Forest cũng được sử dụng rộng rãi trong việc phát hiện gian lận, đặc biệt là các giao dịch trực tuyến. Với các dữ liệu về giao dịch tài chính, Random Forest có thể phát hiện các mẫu gian lận dựa trên các dấu hiệu bất thường và hành vi người dùng.

##### 1.3.3.3 Y học và Chẩn Đoán Hình Ảnh

Trong chẩn đoán y tế, Random Forest được sử dụng để phát hiện các loại bệnh, chẳng hạn như phát hiện ung thư dựa trên các chỉ số sức khỏe và chẩn đoán hình ảnh. Mô hình có khả năng phân loại cao và phát hiện các mẫu bệnh lý tiềm ẩn từ dữ liệu y tế.

#### *1.3.3.4 Các Ứng Dụng Khác*

Random Forest còn được áp dụng trong các lĩnh vực khác như dự báo tài chính, phân tích thị trường, nhận diện giọng nói và xử lý ngôn ngữ tự nhiên, nơi các mô hình cần tính ổn định và khả năng khái quát hóa tốt.

### **1.4 Kết chương**

Random Forest là một trong những mô hình học máy mạnh mẽ và phổ biến nhất, nhờ vào khả năng kết hợp nhiều cây quyết định để đạt độ chính xác và ổn định cao. Mô hình này đặc biệt hiệu quả với các bài toán có dữ liệu lớn, đa dạng và yêu cầu tính chính xác cao, và đã chứng tỏ được hiệu suất xuất sắc trong nhiều lĩnh vực. Trong lĩnh vực an toàn thông tin, Random Forest là một lựa chọn lý tưởng cho việc phân tích log web để phát hiện các cuộc tấn công, nhờ khả năng phân loại chính xác các yêu cầu bất thường mà không đòi hỏi quá nhiều điều chỉnh phức tạp.

Random Forest sẽ được sử dụng trong báo cáo này như một bộ phân loại chính, với mục tiêu phân loại các URI từ log web thành các loại "bình thường" và "tấn công", bao gồm SQL Injection, XSS, Path Traversal, và OS Command Injection.

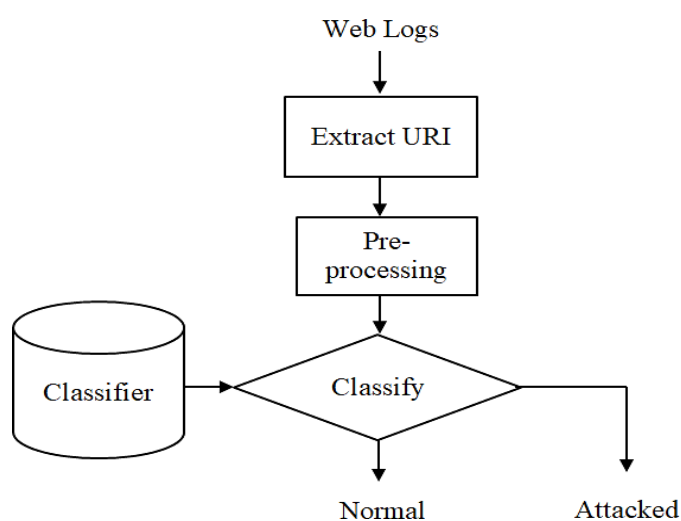
## CHƯƠNG 2. THIẾT KẾ VÀ TRIỂN KHAI MÔ HÌNH PHÁT HIỆN TẤN CÔNG

### 2.1 Phương pháp nghiên cứu

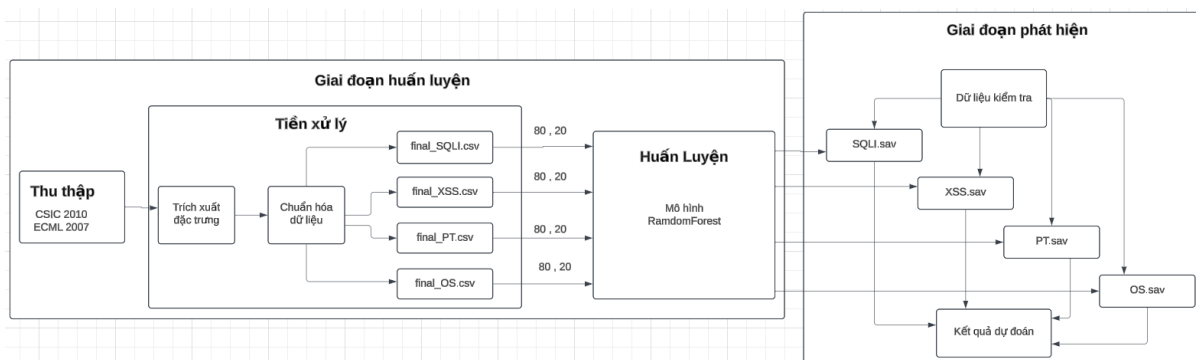
#### 2.1.1 Kiến trúc mô hình phát hiện tấn công

Mô hình phát hiện tấn công web được chia thành hai giai đoạn chính: Giai đoạn Huấn luyện và Giai đoạn Phát hiện. Trong giai đoạn huấn luyện, mô hình sẽ học cách phân biệt giữa các URI bình thường và các URI chứa tấn công từ một bộ dữ liệu huấn luyện có sẵn. Sau khi được huấn luyện, mô hình sẽ được sử dụng trong giai đoạn phát hiện để phân loại các URI mới từ log web, xác định xem chúng có dấu hiệu của các cuộc tấn công hay không.

Sơ đồ tổng quan của quy trình:



Hình ảnh 17: Mô hình đề xuất cho phát hiện tấn công Web



Hình ảnh 18: Mô hình chi tiết của phát hiện tấn công Web

#### 2.1.2 Giai đoạn Huấn luyện

Bước 1: Thu thập dữ liệu huấn luyện

- Nguồn dữ liệu: Trong nghiên cứu này, bộ dữ liệu huấn luyện được lấy từ các nguồn công khai và uy tín là ECML. Bộ dữ liệu này được biết đến với tính phong phú và đa dạng trong các URI chứa các loại tấn công khác nhau, đặc biệt là các dạng tấn công SQL Injection, Cross-Site Scripting (XSS), Path Traversal và OS Command

Injection, đồng thời cũng bao gồm nhiều URI bình thường để đảm bảo tính chính xác cho quá trình huấn luyện.

```
▶ d1 = pd.read_csv("drive/MyDrive/Machine_Log_Analysis/Dataset/ecml/ecml.csv")
```

Hình ảnh 19: Đọc dữ liệu vào dataframe d1

- Mô tả về bộ dữ liệu: Dữ liệu từ ECML cung cấp các mẫu log web chi tiết, giúp mô hình học được đặc điểm của các URI bình thường và các URI tấn công. Bộ dữ liệu này không chỉ giúp mô hình nhận diện chính xác các kiểu tấn công đã biết mà còn hỗ trợ trong việc phát hiện các mẫu tấn công mới dựa trên những điểm tương đồng với các mẫu đã có.

General Info:

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 68269 entries, 0 to 68268
```

```
Data columns (total 15 columns):
```

#	Column	Non-Null Count	Dtype
0	id	68269 non-null	int64
1	ns1:os	68269 non-null	object
2	ns1:webserver	68269 non-null	object
3	ns1:runningLdap	68269 non-null	object
4	ns1:runningSqlDb	68269 non-null	object
5	ns1:runningXpath	68269 non-null	object
6	ns1:type	52296 non-null	object
7	ns1:inContext	68269 non-null	bool
8	ns1:attackIntervall	15973 non-null	object
9	ns1:method	68269 non-null	object
10	ns1:protocol	68269 non-null	object
11	ns1:uri	68269 non-null	object
12	ns1:query	47924 non-null	object
13	ns1:headers	68269 non-null	object
14	ns1:body	12336 non-null	object

```
dtypes: bool(1), int64(1), object(13)
```

```
memory usage: 7.4+ MB
```


```
None
```

Hình ảnh 20: Mô tả tổng quan về nguồn dữ liệu

## Bước 2: Tiền xử lý dữ liệu và trích xuất đặc trưng

- Quá trình tiền xử lý:
  - Chuẩn hóa dữ liệu để đảm bảo tính nhất quán. Các URI sẽ được chuẩn hóa để loại bỏ các yếu tố như dấu phân cách hoặc khoảng trắng không cần thiết.

- Phân tách các phần quan trọng của URI như đường dẫn, và các tham số query, body. Các tham số này thường chứa nhiều thông tin về các kiểu tấn công (ví dụ: chuỗi độc hại trong tấn công SQL Injection hoặc XSS).



	url
type	
OS	3420
PT	3029
SQLi	42032
Valid	43369
XSS	6643

Hình ảnh 21: Xử lý dữ liệu

```

⇒ type Valid
url /l_t@/_Feu1wwhtpass2/1nieQnnrvnzktuasain/tg1AR...
Name: 0, dtype: object
-----
type SQLi
url /cShcktp/WGRj_l3T4VCIAM/t3Ww_4V69aXiVXc6jx/iAd...
Name: 35006, dtype: object
-----
type OS
url /cShcktp/WGRj_l3T4VCIAM/t3Ww_4V69aXiVXc6jx/iAd...
Name: 35007, dtype: object
-----
type PT
url /jF1GYRSYPARBHoPbHj/4AT/b8e4ttsysw/EaweeIs/vxV...
Name: 35501, dtype: object
-----
type XSS
url /<script>alert('Vulnerable')</script>.jsp
Name: 44015, dtype: object
-----
type anomalous
url /AnToanWeb
Name: 45840, dtype: object
-----
type attack
url /antoanweb/publico/autenticar.jsp?login=61%27%...
Name: 55840, dtype: object
-----

```

Hình ảnh 22: Format của dữ liệu

- Chuyển đổi URI thành các vector đặc trưng, có thể bao gồm các yếu tố như độ dài của URI, tần suất xuất hiện của các ký tự đặc biệt (ví dụ: dấu ?, =, %, hoặc &), và các chuỗi nghi ngờ.

df.head()

	Unnamed: 0.1	Unnamed: 0	type	url	label	.bat	log	:	//	winnt	...	./	..\\	file	passwd	:\\	.conf	\\	\\	..\\	:\\
0	0	0	Valid	/L_t@/_feu1wvhtpass2/1nieqnnrvnzkluasain/tg1ar...	0.0	0	0	0	0	0	...	0	0.0	0	0	0.0	0	0	0	0	0
1	1	1	Valid	/inssgtz7ieltssstbw7e/nehqmsdwu7imdb0etet/et/h...	0.0	0	0	0	0	...	0	0.0	0	0	0.0	0	0	0	0	0	0
2	2	2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0.0	0	0	0	0	...	0	0.0	0	0	0.0	0	0	0	0	0	0
3	3	3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0.0	0	0	0	0	...	0	0.0	0	0	0.0	0	0	0	0	0	0
4	4	4	Valid	/2m6vlb1r37jpc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0.0	0	0	0	0	...	0	0.0	0	0	0.0	0	0	0	0	0	0

5 rows x 35 columns

Hình ảnh 23: Trích xuất đặc trưng cho Path traversal

df

	type	url	label	Unnamed: 4	between	like	%	"	table	^	...		exec	delete	select	]	<>	!=	&&	<	any
0	Valid	/L_t@/_feu1wvhtpass2/1nieqnnrvnzkluasain/tg1ar...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	Valid	/inssgtz7ieltssstbw7e/nehqmsdwu7imdb0etet/et/h...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Valid	/2m6vlb1r37jpc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
126146	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126147	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126148	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126149	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126150	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Hình ảnh 24: Trích xuất đặc trưng cho SQLi

df

	type	url	label	Unnamed: 4	&#	#	*	:	,	search	...	<>	;	iframe	(	http	createelement	+	div	'	src
0	Valid	/L_t@/_feu1wvhtpass2/1nieqnnrvnzkluasain/tg1ar...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
1	Valid	/inssgtz7ieltssstbw7e/nehqmsdwu7imdb0etet/et/h...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	Valid	/2m6vlb1r37jpc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
126146	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126147	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126148	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126149	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126150	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

125869 rows x 57 columns

Hình ảnh 25: Trích xuất đặc trưng cho XSS

df

	type	url	label	Unnamed: 4	-aux	access	:	ftp	passwd	..	...	../	http	./	cat	display	bin/	'	.exe	ping	IP
0	Valid	/L_t@/_feu1wvhtpass2/1nieqnnrvnzkluasain/tg1ar...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	Valid	/inssgtz7ieltssstbw7e/nehqmsdwu7imdb0etet/et/h...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Valid	/2m6vlb1r37jpc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3107532	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3107533	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3107534	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3107535	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3107536	Valid	/antoanweb/publico/registro.jsp?modo=registro&...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

125869 rows x 50 columns

Hình ảnh 26: Trích xuất đặc trưng cho OS injection

- Chuyển đổi thành vector và ma trận huấn luyện:
  - Mỗi URI sau khi tiền xử lý sẽ được chuyển đổi thành một vector đặc trưng, bao gồm các giá trị số mô tả tính chất của URI.



- Tập dữ liệu huấn luyện sẽ trở thành một ma trận  $M \times (N + 1)$ , trong đó:
  - $M$  là tổng số URI trong tập huấn luyện.
  - $N$  là số lượng đặc trưng (số lượng cột đặc trưng trong vector của URI).
  - Cột cuối của ma trận này chứa nhãn của URI, chỉ định "bình thường" hoặc "tấn công" (cụ thể là loại tấn công như SQL Injection, XSS, Path Traversal, hoặc OS Command Injection).
- Gắn nhãn dữ liệu:
  - Mỗi URI sẽ được gắn nhãn là "bình thường" hoặc "tấn công". Nếu là URI tấn công, cần chỉ rõ loại tấn công để mô hình có thể phân biệt các loại này.

### Bước 3: Xây dựng mô hình

```
df = pd.read_csv("/content/drive/MyDrive/Machine_Log_Analysis/Dataset/Final_Csv/final_OS.csv")
df = df.dropna()
del df['Unnamed: 0']

df = pd.read_csv("/content/drive/MyDrive/Machine_Log_Analysis/Dataset/Final_Csv/final_SQLi.csv")
df = df.dropna()
del df['Unnamed: 0']

df = pd.read_csv("/content/drive/MyDrive/Machine_Log_Analysis/Dataset/Final_Csv/final_PT.csv")
df = df.dropna()
del df['Unnamed: 0']

df = pd.read_csv("/content/drive/MyDrive/Machine_Log_Analysis/Dataset/Final_Csv/final_XSS.csv")
df = df.dropna()
del df['Unnamed: 0']
```

Hình ảnh 27: Đọc dữ liệu vào dataframe

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import metrics

X = list(df.columns[3:])
X_df = df[X]
Y_df = df['label']
X_train, X_test, Y_train, Y_test = train_test_split(X_df, Y_df, test_size=0.2, random_state=0)
```

Hình ảnh 28: Chia tập dữ liệu để huấn luyện

- Huấn luyện mô hình Random Forest:
  - o Tập dữ liệu đã được chuyển đổi và gắn nhãn sẽ được sử dụng để huấn luyện mô hình Random Forest.
  - o Các tham số quan trọng của mô hình:
    - Số lượng cây (estimators): Quyết định số cây được sử dụng trong mô hình Random Forest. Số lượng cây lớn có thể tăng độ chính xác nhưng sẽ tăng thời gian huấn luyện.
    - Độ sâu tối đa của cây (max depth): Quy định độ sâu tối đa mà mỗi cây quyết định có thể phát triển. Giới hạn độ sâu giúp giảm thiểu nguy cơ overfitting.
    - Các tham số khác như min\_samples\_split, min\_samples\_leaf giúp tinh chỉnh mô hình cho hiệu quả cao hơn.

- Sau quá trình huấn luyện, mô hình Random Forest sẽ được tối ưu hóa để phân loại các URI thành các nhóm "bình thường" hoặc các nhóm tấn công.

```

▶ from sklearn.ensemble import RandomForestClassifier

# Khởi tạo mô hình với các tham số đã chọn
RF = RandomForestClassifier(
    n_estimators=100, # Số lượng cây
    max_depth=10,     # Độ sâu tối đa
    min_samples_split=2, # Số mẫu tối thiểu để chia
    min_samples_leaf=1, # Số mẫu tối thiểu tại lá
    max_features='sqrt', # Số lượng tính năng tối đa
    bootstrap=True,     # Sử dụng sampling với thay thế
    class_weight='balanced' # Trọng số lớp
)

# Huấn luyện mô hình
RF.fit(X_train, Y_train)

```

Hình ảnh 29: Xây dựng tham số cho Random Forest

#### Bước 4: Lưu mô hình

```

import pickle
filename = '/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results/OS.sav'
pickle.dump(GB, open(filename, 'wb'))
#loaded_model = pickle.load(open(filename, 'rb'))
#result = loaded_model.score(X_test, Y_test)
#print(result)
# !cp OS.sav "/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results"

▶ import pickle
filename = '/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results/XSS.sav'
pickle.dump(GB, open(filename, 'wb'))
#loaded_model = pickle.load(open(filename, 'rb'))
#result = loaded_model.score(X_test, Y_test)
#print(result)
# !cp OS.sav "/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results"

▶ import pickle
filename = '/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results/SQLi.sav'
pickle.dump(GB, open(filename, 'wb'))
#loaded_model = pickle.load(open(filename, 'rb'))
#result = loaded_model.score(X_test, Y_test)
#print(result)
# !cp OS.sav "/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results"

▶ import pickle
filename = '/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results/PT.sav'
pickle.dump(GB, open(filename, 'wb'))
#loaded_model = pickle.load(open(filename, 'rb'))
#result = loaded_model.score(X_test, Y_test)
#print(result)
# !cp OS.sav "/content/drive/MyDrive/Machine_Log_Analysis/Model/Model_Results"

```

Hình ảnh 30: Lưu mô hình đã huấn luyện

### 2.1.3 Giai đoạn Phát hiện

#### Bước 1: Tiền xử lý URI trong log web

- Trích xuất và chuẩn hóa URI: Trong giai đoạn phát hiện, các URI sẽ được trích xuất từ log web của ứng dụng. Các log này ghi lại toàn bộ hoạt động truy cập và yêu cầu từ người dùng, giúp phát hiện các hành vi bất thường.

- Xử lý URI thành vector đặc trưng: Quá trình này tương tự như giai đoạn huấn luyện. Các URI sẽ được chuyển đổi thành vector đặc trưng theo cùng cách thức để đảm bảo tính nhất quán giữa giai đoạn huấn luyện và phát hiện. Điều này bao gồm các bước chuẩn hóa, phân tích cấu trúc URI, và trích xuất đặc trưng.

## Bước 2: Phân loại URI

- Sử dụng mô hình Random Forest để phân loại: Mô hình Random Forest đã được huấn luyện sẽ nhận các vector đặc trưng từ URI mới và phân loại chúng thành "bình thường" hoặc "tấn công".
- Xác định trạng thái của URI: Nếu URI được phân loại là tấn công, mô hình sẽ xác định loại tấn công cụ thể (SQL Injection, XSS, Path Traversal, hoặc OS Command Injection).
- Kết quả: Mỗi URI sẽ được gán nhãn và trạng thái của nó sẽ được cập nhật trong hệ thống. Các URI được phát hiện là tấn công sẽ được ghi nhận, và các biện pháp phòng ngừa sẽ được thực hiện để ngăn chặn các hành vi tương tự trong tương lai.

*Với kiến trúc này, mô hình sẽ có khả năng phát hiện các tấn công từ log web và hỗ trợ các tổ chức trong việc bảo vệ ứng dụng khỏi các mối đe dọa bảo mật*

## 2.2 Kết chương

Trong chương này, chúng ta đã trình bày một phương pháp thiết kế và triển khai mô hình phát hiện tấn công cho các ứng dụng web, đặc biệt là trong việc nhận diện các kiểu tấn công như SQL Injection, XSS, Path Traversal và OS Command Injection. Mô hình được chia thành hai giai đoạn chính: Giai đoạn Huấn luyện và Giai đoạn Phát hiện.

Trong giai đoạn huấn luyện, mô hình sử dụng một bộ dữ liệu huấn luyện phong phú và đa dạng từ nguồn ECML để học cách phân biệt giữa các URI bình thường và các URI chứa tấn công. Các bước tiền xử lý dữ liệu, trích xuất đặc trưng và xây dựng mô hình Random Forest đã được thực hiện để tối ưu hóa hiệu quả phát hiện. Kết quả huấn luyện đã cho phép mô hình phân loại chính xác các URI và nhận diện các hành vi tấn công.

Giai đoạn phát hiện là quá trình tiếp theo, trong đó mô hình đã được áp dụng để phân loại các URI mới trong log web của hệ thống. Các URI được trích xuất, chuẩn hóa và chuyển đổi thành các vector đặc trưng, từ đó mô hình Random Forest sẽ phân loại chúng là "bình thường" hoặc "tấn công". Những URI chứa tấn công sẽ được ghi nhận và cảnh báo, giúp hệ thống ứng dụng có thể phản ứng kịp thời và giảm thiểu các mối đe dọa bảo mật.

Với mô hình này, chúng ta không chỉ có khả năng phát hiện các cuộc tấn công đã biết mà còn có thể ứng dụng trong việc nhận diện các tấn công mới dựa trên các đặc trưng đã học được từ dữ liệu huấn luyện. Qua đó, mô hình góp phần nâng cao mức độ an toàn và bảo mật cho các ứng dụng web trong môi trường internet ngày nay.

## CHƯƠNG 3. THỬ NGHIỆM, ĐÁNH GIÁ VÀ ỨNG DỤNG THỰC TIỄN

### 3.1 Kết quả thực nghiệm

Để đánh giá hiệu suất của mô hình Random Forest trong phát hiện các cuộc tấn công trên log web, chúng ta sử dụng một số chỉ số chính gồm:

- Accuracy: Tỷ lệ các URI (bình thường và tấn công) mà mô hình phân loại đúng trên tổng số URI.
- Precision: Đo lường tỷ lệ chính xác khi mô hình phân loại URI là tấn công, tức là tỷ lệ URI thực sự là tấn công trong số các URI được phân loại là tấn công.
- Recall: Đo lường khả năng phát hiện các URI tấn công của mô hình trong tất cả các URI thực sự là tấn công.
- F1-score: Là trung bình điều hòa giữa Precision và Recall, cung cấp một cái nhìn cân bằng hơn về hiệu suất của mô hình, đặc biệt khi dữ liệu mất cân bằng (số lượng URI bình thường nhiều hơn số lượng URI tấn công).

#### 3.1.1 Hiệu suất mô hình

##### 3.1.1.1 Accuracy (Độ chính xác)

Công thức:

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

Hình ảnh 31: Công thức tính Accuracy

- Accuracy là tỷ lệ số lượng URI được phân loại đúng (gồm cả URI bình thường và URI tấn công) trên tổng số URI.
- Ở đây:
  - TP (True Positive): Số lượng URI tấn công được mô hình phân loại đúng là tấn công.
  - TN (True Negative): Số lượng URI bình thường được mô hình phân loại đúng là bình thường.
  - FP (False Positive): Số lượng URI bình thường nhưng lại bị phân loại sai là tấn công.
  - FN (False Negative): Số lượng URI tấn công nhưng lại bị phân loại sai là bình thường.

*Accuracy cung cấp cái nhìn tổng quan về tỷ lệ phân loại đúng của mô hình trên toàn bộ dữ liệu. Tuy nhiên, khi dữ liệu mất cân bằng (số lượng URI bình thường nhiều hơn URI tấn công), độ chính xác có thể bị lệch.*

##### 3.1.1.2 Precision (Độ chính xác dự đoán)

Công thức:

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

Hình ảnh 32: Công thức tính Precision

- Precision đo lường tỷ lệ các URI thực sự là tấn công trong số các URI mà mô hình đã phân loại là tấn công.
- Precision giúp đánh giá độ chính xác của mô hình khi xác định các URI tấn công, giảm thiểu số lượng cảnh báo sai (False Positive).

Precision càng cao thì mô hình càng ít báo động nhầm (FP) và có tính chính xác cao hơn khi dự đoán URI là tấn công.

#### 3.1.1.3 Recall (Độ phủ)

Công thức:

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

Hình ảnh 33: Công thức tính Recall

- Recall đo lường khả năng của mô hình trong việc phát hiện đúng các URI tấn công trên tổng số URI thực sự là tấn công.
- Chỉ số này giúp đánh giá mức độ bao phủ của mô hình, tức là khả năng không bỏ sót các URI tấn công (giảm thiểu False Negative).

Recall cao đảm bảo rằng mô hình có thể phát hiện được hầu hết các URI tấn công.

#### 3.1.1.4 F1-score

Công thức:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Hình ảnh 34: Công thức tính F1-score



- F1-score là trung bình điều hòa giữa Precision và Recall, cung cấp một cái nhìn cân bằng giữa hai chỉ số này.
- F1-score đặc biệt hữu ích khi dữ liệu mất cân bằng, vì nó giúp đánh giá mô hình không chỉ dựa vào một trong hai yếu tố Precision hay Recall, mà là sự kết hợp của cả hai.

F1-score cao nghĩa là mô hình có khả năng dự đoán chính xác URI tấn công (Precision cao) và đồng thời có khả năng phát hiện hầu hết các URI tấn công (Recall cao).

### 3.1.2 Kết quả dự đoán thực nghiệm và hiệu suất của mô hình

#### 3.1.2.1 Kết quả dự đoán thực nghiệm

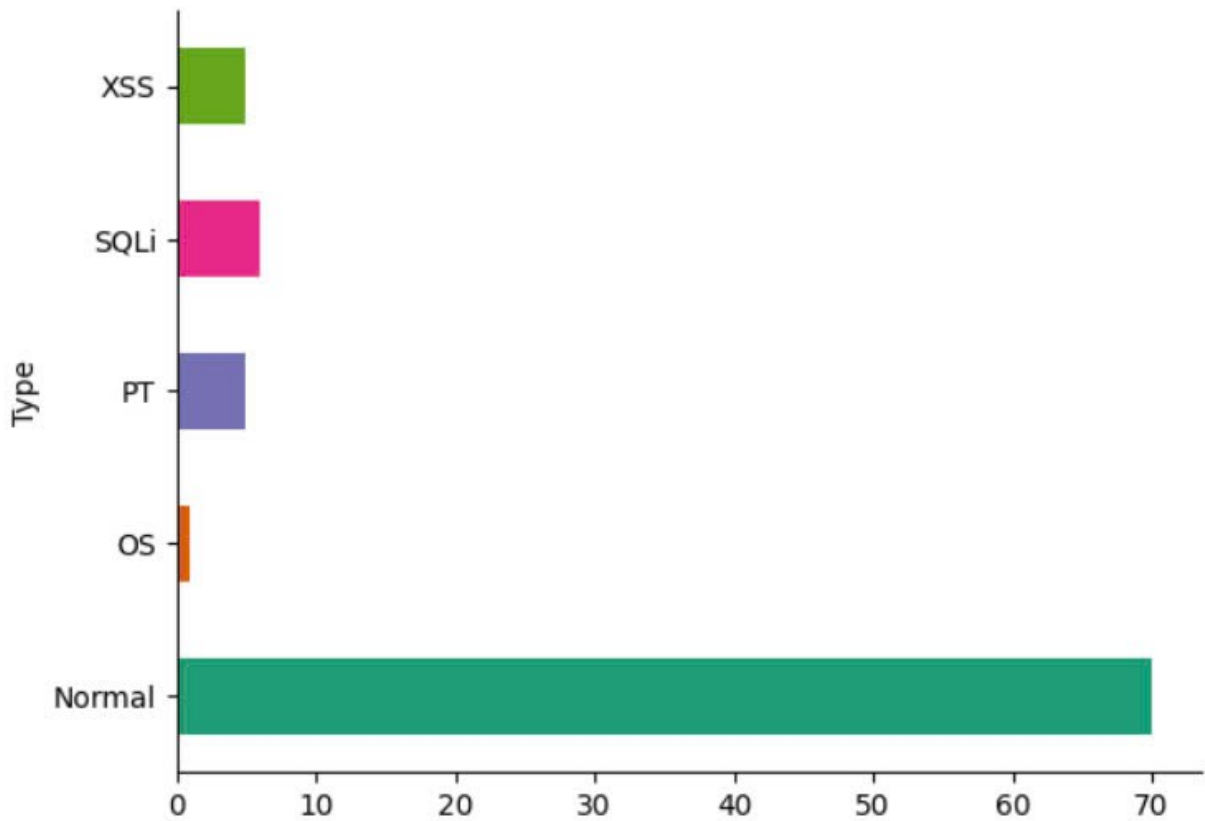
	IP Address	Timestamp	Method	Request Vector	Response Code	Type
0	127.0.0.1	16/Dec/2020:13:55:39 +0530	GET	/sbva/assets/fonts/glyphicons-halflings-regula...	404	Normal
1	127.0.0.1	16/Dec/2020:13:55:39 +0530	GET	/sbva/assets/fonts/glyphicons-halflings-regula...	404	Normal
2	127.0.0.1	16/Dec/2020:13:55:39 +0530	GET	/sbva/assets/fonts/glyphicons-halflings-regula...	404	Normal
3	127.0.0.1	16/Dec/2020:13:55:42 +0530	GET	/sbva/contact.php	200	Normal
4	127.0.0.1	16/Dec/2020:13:56:11 +0530	POST	/sbva/contact.php	200	Normal
...	...	...	...	...	...	...
82	192.168.199.2	20/Jun/2021:12:36:33 +0300	GET	/bwapp/upload.php?type=\\<script>alert(docum...	404	XSS
83	192.168.199.2	20/Jun/2021:12:36:33 +0300	GET	/bwapp/addyoursite.php?catid=&lt;Script&gt;Jav...	404	Normal
84	192.168.199.2	20/Jun/2021:12:36:34 +0300	GET	/bwapp/myphpnuke/links.php?op=search&query=[sc...	404	XSS
85	192.168.199.2	20/Jun/2021:12:36:33 +0300	GET	/bwapp/postnuke/html/index.php?module=My_eGall...	404	SQLi
86	192.168.199.2	20/Jun/2021:12:36:33 +0300	GET	/bwapp/index.php?module=My_eGallery&do=showpic...	302	SQLi

87 rows x 6 columns

Hình ảnh 35: Kết quả dự đoán

	A	B	C	D	E	F
66	127.0.0.1	16/Dec/2020:	GET	/dvwa/vulnerabilities/sqli/?id=SELECT+*+FROM+Users+V	200	Normal
67	127.0.0.1	18/Dec/2020:	GET	/dvwa/vulnerabilities/sqli/?id=SELECT+*+FROM+Users+V	200	SQLi
68	192.168.199.2	20/Jun/2021:	GET	/bwapp/help/../../../../../../../../etc/shadow	400	PT
69	192.168.199.2	20/Jun/2021:	GET	/bwapp/xsql/demo/adhocsql/query.xsql?sql=select%20use	404	SQLi
70	192.168.199.2	20/Jun/2021:	GET	/bwapp/coast/header.php?sections_file=http://cirt.net/rfiinc	404	PT
71	192.168.199.2	20/Jun/2021:	GET	/bwapp/theme1/selector?button=status,monitor,session&b	404	XSS
72	127.0.0.1	18/Dec/2020:	GET	/dvwa/vulnerabilities/exec/	200	Normal
73	192.168.199.2	20/Jun/2021:	GET	/bwapp/error/cat\${IFS}/etc/passwd	404	OS
74	127.0.0.1	18/Dec/2020:	POST	/dvwa/vulnerabilities/exec/	200	Normal
75	127.0.0.1	18/Dec/2020:	POST	/dvwa/vulnerabilities/exec/	200	Normal
76	127.0.0.1	18/Dec/2020:	POST	/dvwa/vulnerabilities/exec/system('ls')	200	Normal
77	127.0.0.1	18/Dec/2020:	GET	/dvwa/vulnerabilities/csrf/	200	Normal
78	192.168.199.2	20/Jun/2021:	GET	/bwapp/forum/mail.php?repertorylevel=http://cirt.net/rfiinc.t	404	Normal
79	192.168.199.2	20/Jun/2021:	GET	/bwapp/forum/message.php?repertorylevel=http://cirt.net/r	404	Normal
80	127.0.0.1	18/Dec/2020:	GET	/dvwa/vulnerabilities/sqli/?id=%22SELECT+*+FROM+Use	200	SQLi
81	192.168.199.2	20/Jun/2021:	GET	/bwapp/cmsimple2_7/cmsimple/cms.php?pth[file]['config']	404	PT
82	192.168.199.2	20/Jun/2021:	GET	/bwapp/forum.asp?n=../../../../../../../../etc/passwd%00 41	400	PT
83	127.0.0.1	18/Dec/2020:	GET	/dvwa/vulnerabilities/sqli/?id=INSERT+INTO+Customers+'	200	SQLi
84	192.168.199.2	20/Jun/2021:	GET	/bwapp/upload.php?type=\\<script>alert(document.cookie	404	XSS

Hình ảnh 36: Kết quả dự đoán ở dạng file sheet



Hình ảnh 37: Tỷ lệ giữa các hành vi bình thường và các cuộc tấn công

### 3.1.2.2 Hiệu suất mô hình

- Accuracy: Đánh giá tổng thể nhưng có thể không phản ánh đúng nếu dữ liệu mất cân bằng.
- Precision: Quan trọng khi cần giảm cảnh báo sai (False Positive), giúp mô hình chính xác hơn trong các dự đoán tấn công.
- Recall: Phù hợp khi ưu tiên phát hiện tối đa các URI tấn công, tránh bỏ sót.
- F1-score: Kết hợp cả Precision và Recall, giúp đánh giá cân bằng hơn khi dữ liệu mất cân bằng giữa URI bình thường và URI tấn công.

Kết quả cụ thể về hiệu suất của mô hình Random Forest:

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import ConfusionMatrixDisplay
#Tạo model RandomForest RF và bắt đầu huấn luyện
RF = RandomForestClassifier()
RF.fit(X_train,Y_train)

print('Độ chính xác của mô hình phân loại Random Forest trên tập huấn luyện đối với XSS: {:.2f}'
      .format(RF.score(X_train, Y_train)))
print('Độ chính xác của mô hình phân loại Random Forest trên tập kiểm tra đối với XSS: {:.2f}'
      .format(RF.score(X_test, Y_test)))

#In ra kết quả huấn luyện
print()
Y_pred = RF.predict(X_test)

# In ra ma trận nhầm lẫn
cm = confusion_matrix(Y_test, Y_pred)

ConfusionMatrixDisplay(confusion_matrix=cm).plot();
metrics.accuracy_score(Y_test, Y_pred)
# In ra độ chính xác và các chỉ số
accuracy = accuracy_score(Y_test, Y_pred)
precision = precision_score(Y_test, Y_pred)
recall = recall_score(Y_test, Y_pred)
F1_score = f1_score(Y_test,Y_pred)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1_score",F1_score);

```

Hình ảnh 38: Sử dụng mô hình và đánh giá kết quả

Chỉ số	SQL Injection	XSS	Path Traversal	OS Command Injection
Accuracy	97.56%	99.6%	98.8%	98.22%
Precision	96.6%	99.4%	85.26%	81.15%
Recall	96.5%	93.3%	60.54%	47.6%
F1-score	96.6%	96.2%	70.81%	60.05%

Bảng 1: Đánh giá trên từng loại tấn công

Các chỉ số này cho thấy rằng mô hình Random Forest đạt được độ chính xác cao trong việc phân loại SQL injection và XSS và có chỉ số F1-score cao cân bằng tốt giữa precision và recall .

Nhưng trong việc phát hiện các loại tấn công Path Traversal và OS Command Injection , với chỉ số F1-score là 70.81% và 60.05\$ , chứng tỏ chưa cân bằng tốt giữa precision và recall.

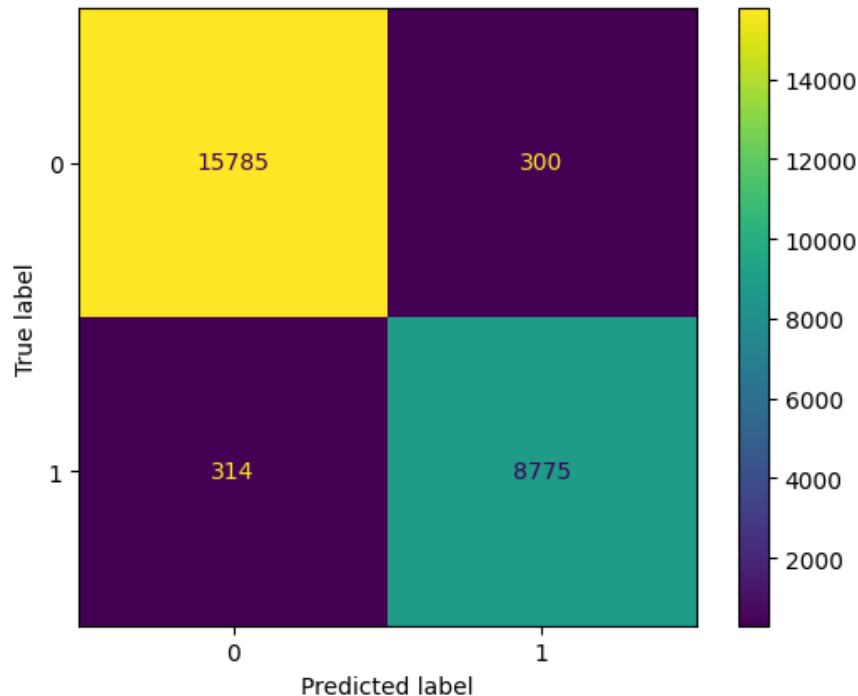
### 3.1.3 Phân tích hiệu suất mô hình theo từng loại tấn công

#### 3.1.3.1 SQL injection



Accuracy of Random Forest classifier on training set of SQLi: 0.98  
 Accuracy of Random Forest classifier on test set of SQLi: 0.98

Accuracy: 0.975609756097561  
 Precision: 0.9669421487603306  
 Recall: 0.9654527450764661  
 F1\_score 0.9661968729354767



Hình ảnh 39: Hiệu suất mô hình đối với SQL injection

- **Độ chính xác**

- ⇒ Độ chính xác của tập dữ liệu huấn luyện là **98 %**.
- ⇒ Độ chính xác của tập dữ liệu kiểm tra là **98%**.
- ⇒ **Độ chính xác tổng thể là 97.5%**

**Nhận xét:** Tập huấn luyện và kiểm tra đều đạt độ chính xác 98%, điều này cho thấy mô hình không bị overfitting. Mô hình đang hoạt động tốt và có khả năng tổng quát hóa cao trên dữ liệu chưa từng gặp.

- **Ma trận nhầm lẫn**

Lớp 0 – NonSQLi:

- 15755 số lượng mẫu Non-SQLi được dự đoán đúng là Non-SQLi (True Negatives – TN ).
- 330 số lượng mẫu Non-SQLi bị dự đoán nhầm là SQLi (False Positive -FP).

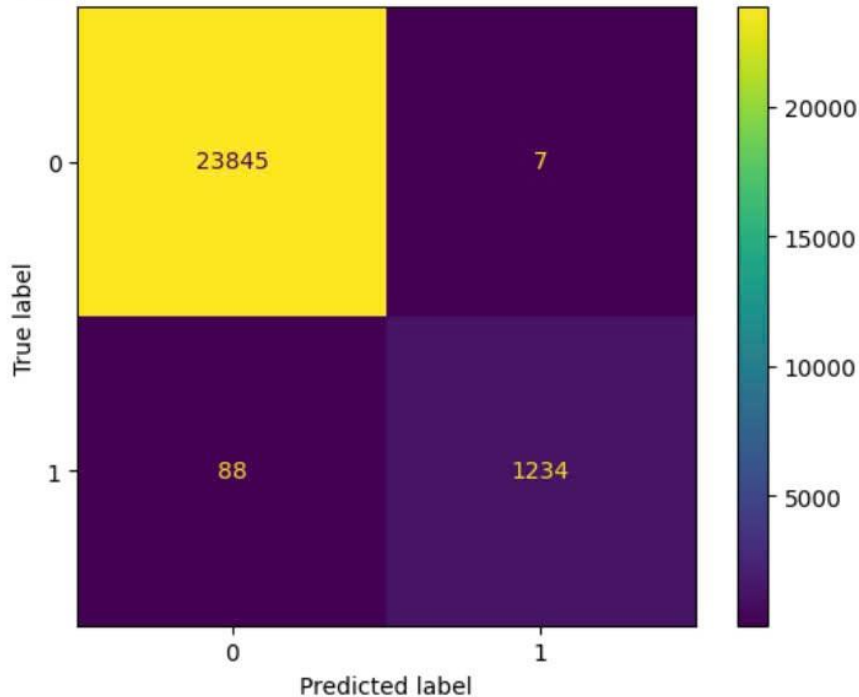
Lớp 1 – SQLi:

- 8799 số lượng mẫu SQLi bị dự đoán đúng là SQLi (True Positives -TP).
- 290 số lượng mẫu SQLi bị dự đoán nhầm là Non-SQLi (False Negatives - FN).

### 3.1.3.2 XSS

Accuracy of Random Forest classifier on training set of XSS: 1.00  
Accuracy of Random Forest classifier on test set of XSS: 1.00

Accuracy: 0.9962262651942481  
Precision: 0.9943593875906527  
Recall: 0.9334341906202723  
F1\_score 0.962934061646508



Hình ảnh 40: Hiệu suất mô hình đối với XSS

- **Độ chính xác**

- ⇒ Độ chính xác của tập dữ liệu huấn luyện là **100 %**.
- ⇒ Độ chính xác của tập dữ liệu kiểm tra là **100%**.
- ⇒ **Độ chính xác tổng thể là 99.62 %**

**Nhận xét:** Mô hình Random Forest đạt độ chính xác tuyệt đối (100%) trên tập huấn luyện, cho thấy nó đã học được hoàn toàn các đặc trưng của dữ liệu huấn luyện. Trên tập kiểm tra độ chính xác gần như tuyệt đối (99.6%) cho thấy mô hình tổng quát hóa rất tốt trên dữ liệu chưa từng thấy. Mô hình này không bị hiện tượng overfitting (vì accuracy trên test set cũng rất cao).

- **Ma trận nhầm lẫn**

Lớp 0 – NonXSS:

- 23845 số lượng mẫu Non-XSS được dự đoán đúng là Non-XSS (True Negatives – TN).
- 7 số lượng mẫu Non-XSS bị dự đoán nhầm là XSS (False Positive -FP).

Lớp 1 – XSS:

- 1234 số lượng mẫu XSS được dự đoán đúng là XSS (True Positives -TP).
- 88 số lượng mẫu XSS bị dự đoán nhầm là Non-XSS (False Negatives - FN).

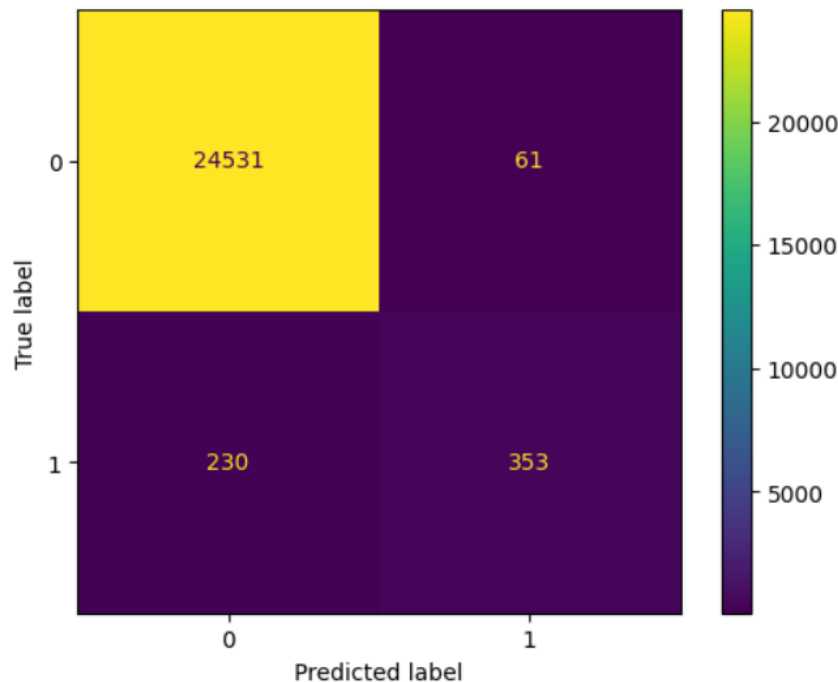
**Kết luận:** Mô hình đạt hiệu suất rất cao trong việc phân loại **Non-XSS** với các chỉ số Precision, Recall, và F1-Score đều đạt 1.00, cho thấy khả năng nhận diện chính xác và đầy đủ các mẫu không phải XSS. Tuy nhiên, với lớp **XSS**, mô hình vẫn hoạt động tốt (Precision 0.99 và F1-Score 0.96) nhưng chưa hoàn hảo, do tỷ lệ phát hiện (Recall) chỉ đạt 0.93. Điều này có nghĩa là mô hình

có nguy cơ bỏ sót một số mẫu XSS thực tế, cần cải thiện để giảm thiểu các trường hợp bị bỏ sót (False Negatives).

### 3.1.3.3 Path Traversal

Accuracy of Random Forest classifier on training set of PT: 0.99  
Accuracy of Random Forest classifier on test set of PT: 0.99

Accuracy: 0.9884409136047666  
Precision: 0.8526570048309179  
Recall: 0.6054888507718696  
F1\_score 0.708124373119358



Hình ảnh 41: Hiệu suất mô hình đối với Path Traversal

#### - Độ chính xác

- ⇒ Độ chính xác của tập dữ liệu huấn luyện là **99 %**.
- ⇒ Độ chính xác của tập dữ liệu kiểm tra là **99%**.
- ⇒ **Độ chính xác tổng thể là 98.84 %**

**Nhận xét:** Mô hình Random Forest đạt độ chính xác rất cao (99%) trên cả tập huấn luyện và tập kiểm tra, chứng tỏ mô hình đã học được đầy đủ các đặc trưng của dữ liệu và tổng quát hóa tốt trên dữ liệu mới. Không có dấu hiệu của hiện tượng overfitting, vì độ chính xác trên cả hai tập dữ liệu gần như tương đương.

#### - Ma trận nhầm lẫn

Lớp 0 – Negative:

- Số mẫu nonPT được dự đoán đúng là 24531 (True Negative).
- Số mẫu nonPT dự đoán sai là PT là 61 (False Positive)

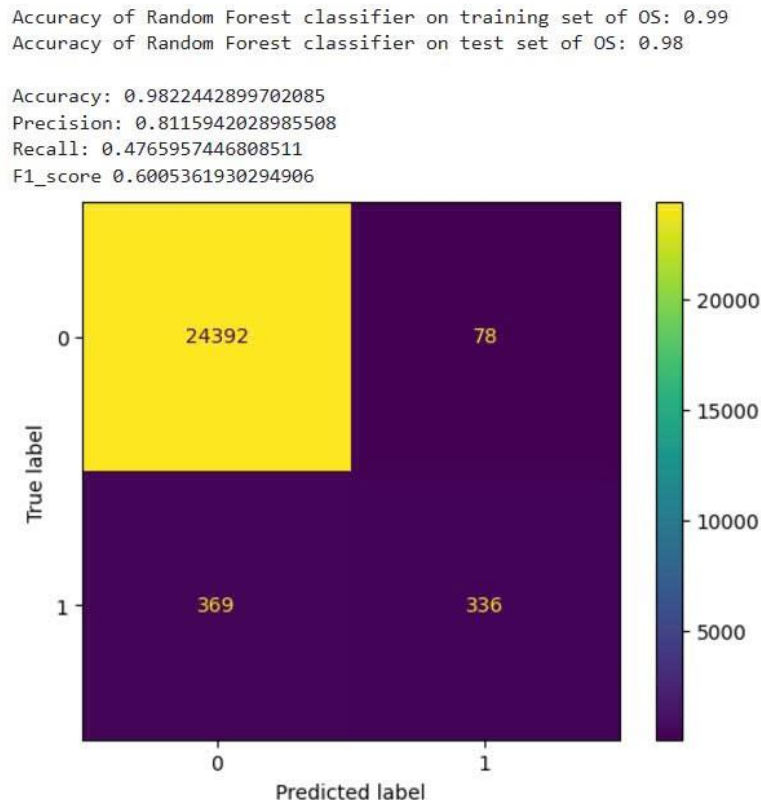
Lớp 1 – Positive:

- Số mẫu PT được dự đoán đúng là 230 (True Positive)
- Số mẫu PT được dự đoán sai là nonPT là 353 (False Positive)

Bảng 2: Chỉ số Precision, Recall, F1-score của Path Traversal

**Kết luận:** Các chỉ số Precision, Recall và F1-Score cao trên cả hai lớp cho thấy mô hình cân bằng và đáng tin cậy.

#### 3.1.3.4 OS Command Injection



Hình ảnh 42: Hiệu suất mô hình đối với OS Command Injection

##### - Độ chính xác

- ⇒ Độ chính xác của tập dữ liệu huấn luyện là **99 %**.
- ⇒ Độ chính xác của tập dữ liệu kiểm tra là **98%**.
- ⇒ **Độ chính xác tổng thể là 0.9822**

**Nhận xét:** Mô hình đạt độ chính xác 99 % cho thấy khả năng học tốt từ dữ liệu huấn luyện và độ chính xác trên tập kiểm tra cũng cao là 98% cho thấy mô hình đang hoạt động ổn định.

##### - Ma trận nhầm lẫn

Lớp 0 – Negative:

- Số mẫu nonOS được dự đoán đúng: 24349 (True Negative).
- Số mẫu là nonOS nhưng dự đoán là OS: 78 (False Positive)

Lớp 1 – Positive:

- Số mẫu OS được dự đoán sai thành nonOS là 369 (False Positive)
- Số mẫu OS được dự đoán đúng :336 (True Positive)

## 3.2 Đánh giá và Thảo luận

### 3.2.1 Ưu điểm và nhược điểm của mô hình

Ưu điểm của Random Forest:

- Hiệu suất cao và khả năng tổng quát tốt: Random Forest là mô hình học máy mạnh mẽ trong xử lý dữ liệu phức tạp với nhiều đặc trưng. Nó có khả năng dự đoán tốt, đồng thời giảm thiểu hiện tượng overfitting (quá khớp) nhờ cơ chế xây dựng nhiều cây quyết định và tổng hợp kết quả.
- Khả năng phát hiện các mẫu chưa từng biết: Với việc huấn luyện trên nhiều đặc trưng của URI, Random Forest có thể nhận diện được các mẫu tấn công mới, không cần phụ thuộc vào các chữ ký đã biết, giúp mô hình có tính linh hoạt cao khi xử lý các tấn công đa dạng.
- Hiệu quả với dữ liệu mất cân bằng: Do cấu trúc của Random Forest với nhiều cây quyết định, mô hình có khả năng xử lý tốt các lớp tấn công dù dữ liệu có sự mất cân bằng, nhờ vào việc phân tích dựa trên nhiều tập con của dữ liệu huấn luyện.

Nhược điểm của Random Forest:

- False Positives: Trong một số trường hợp, mô hình có thể sinh ra nhiều cảnh báo sai (false positives), đặc biệt với các URI có ký tự hoặc mẫu gần giống với các chuỗi tấn công. Điều này đòi hỏi việc hiệu chỉnh mô hình cẩn thận, nhưng vẫn có thể gây khó khăn trong môi trường thực tế nếu số lượng cảnh báo sai quá cao.
- Yêu cầu tài nguyên: Random Forest là một mô hình phức tạp, đòi hỏi tài nguyên tính toán lớn để xây dựng và duy trì hiệu suất cao, đặc biệt khi sử dụng với các bộ dữ liệu lớn hoặc trong thời gian thực. Điều này có thể gây khó khăn nếu áp dụng mô hình trong môi trường có tài nguyên hạn chế.

*Thiếu khả năng giải thích trực quan: So với các mô hình đơn giản hơn như Logistic Regression hay Decision Tree, Random Forest có cấu trúc phức tạp, gây khó khăn trong việc phân tích chi tiết và giải thích quyết định phân loại của mô hình.*

### **3.2.2 Khả năng mở rộng và áp dụng thực tiễn**

Khả năng triển khai thực tế: Mô hình Random Forest có thể được triển khai trong các hệ thống phát hiện và phòng chống xâm nhập (IDS/IPS) để tự động phân loại URI từ log web và phát hiện các tấn công. Tuy nhiên, để triển khai trong môi trường sản xuất, cần tối ưu hóa mô hình để giảm thiểu thời gian xử lý và giảm tỷ lệ cảnh báo sai. Cũng cần đảm bảo rằng mô hình có thể cập nhật dễ dàng với các mẫu dữ liệu mới để duy trì hiệu suất ổn định.

Cải tiến tiềm năng:

- Feature engineering: Nghiên cứu thêm các đặc trưng khác của URI và các thông tin bổ sung từ log (như thông tin tiêu đề HTTP, địa chỉ IP nguồn) có thể giúp cải thiện hiệu suất của mô hình.
- Kết hợp với các mô hình học máy khác: Có thể thử nghiệm kết hợp Random Forest với các mô hình học sâu hoặc các phương pháp ensemble khác để tăng cường khả năng nhận diện và giảm thiểu cảnh báo sai.
- Cập nhật thường xuyên: Với các loại tấn công mới xuất hiện, việc cập nhật mô hình bằng cách huấn luyện trên các bộ dữ liệu mới là cần thiết để tăng khả năng phát hiện của mô hình.

### **3.2.3 Giới hạn của nghiên cứu**

- Giới hạn về dữ liệu: Mặc dù bộ dữ liệu ECML được sử dụng làm nguồn dữ liệu chính, chúng có thể chưa đầy đủ trong việc bao quát toàn bộ các dạng tấn công. Nếu chỉ có một số mẫu URI tấn công, mô hình có thể chưa nhận diện tốt các tấn công ít gặp.
- Hiệu suất trên các loại tấn công ít gặp: Với các tấn công ít gặp hoặc có đặc trưng không rõ ràng như OS Command Injection, khả năng phát hiện của mô hình có thể bị giảm do không có đủ mẫu tấn công để mô hình học tập.
- Tác động của tiền xử lý: Việc chuẩn hóa và chuyển đổi URI có thể làm mất đi một số thông tin ban đầu của URI. Một số đặc trưng quan trọng có thể bị bỏ qua trong quá trình tiền xử lý, làm giảm khả năng phát hiện tấn công.

### 3.3 Kết chương

Chương này đã trình bày và đánh giá hiệu quả của mô hình Random Forest trong việc phát hiện các cuộc tấn công trên log web. Qua kết quả thực nghiệm, mô hình thể hiện khả năng phân loại tốt với độ chính xác cao trong việc nhận diện các cuộc tấn công phổ biến như SQL Injection và XSS. Những chỉ số như Precision, Recall, và F1-score đều đạt mức độ chấp nhận được, cho thấy Random Forest là một công cụ mạnh mẽ trong bảo mật web.

Tuy nhiên, dù mô hình này mang lại hiệu quả cao, nhưng vẫn có một số hạn chế, chẳng hạn như khả năng phát sinh cảnh báo sai (False Positives) hoặc yêu cầu tài nguyên tính toán lớn. Do đó, việc tối ưu hóa và cải thiện mô hình vẫn là một thách thức, đặc biệt là trong việc giảm thiểu các cảnh báo sai và cải thiện độ chính xác chung.

Nhìn chung, Random Forest là một lựa chọn hiệu quả cho việc phát hiện các tấn công mạng, nhưng để ứng dụng thực tế đạt hiệu quả tối ưu, cần tiếp tục nghiên cứu và cải thiện mô hình, đồng thời kết hợp với các phương pháp học máy khác để nâng cao hiệu suất và giảm thiểu nhược điểm.

## KẾT LUẬN

Qua nghiên cứu, mô hình Random Forest cho thấy tiềm năng cao trong việc phát hiện các tấn công web phổ biến như SQL Injection, XSS, Path Traversal và OS Command Injection từ log web. Kết quả thử nghiệm cho thấy mô hình đạt hiệu suất cao với độ chính xác khoảng 96.5% - 99 % cùng với các chỉ số Precision, Recall và F1-score cho thấy mô hình hoạt động tốt và có tính ổn định trong việc phân loại URI .

Nghiên cứu đã chứng minh rằng việc áp dụng machine learning có thể mang lại hiệu quả trong phát hiện tấn công web, giúp giảm thiểu sự phụ thuộc vào các kỹ thuật dựa trên chữ ký truyền thống và cải thiện khả năng phát hiện các tấn công mới Hướng phát triển (nêu hướng phát triển, bổ sung, nghiên cứu tiếp của BTL).

Tuy nhiên, để đạt được hiệu quả tối đa, việc triển khai học máy đòi hỏi cơ sở dữ liệu log chất lượng, kỹ thuật xử lý tiền đề phù hợp và sự phối hợp giữa chuyên gia bảo mật cùng nhà phát triển. Trong tương lai, kết hợp học máy với các công nghệ khác như trí tuệ nhân tạo và học sâu sẽ mở ra những khả năng mới trong việc bảo vệ hệ thống trước các nguy cơ an ninh ngày càng gia tăng.

## TÀI LIỆU THAM KHẢO

- [1] (T. Saranya et al., 2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review
- [2] Hoàng Xuân Dâu. BÀI GIẢNG AN TOÀN ỨNG DỤNG WEB VÀ CƠ SỞ DỮ LIỆU
- [3] [Web Application Security, Testing, & Scanning - PortSwigger](#)
- [4] [OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation](#)