

o o o o

Phân tích log phát hiện tấn công mạng sử dụng học máy

SQL Injection
XSS
Path Traversal
OS Command

Thành viên

◦ ◦ ◦ ◦

Phạm Lê Hoàng Anh

- B21DCAT035

Lý Quốc Khánh

- B21DCAT

Trần Trọng Mạnh

- B21DCAT

Bùi Duy Thanh

- B21DCAT

Nội dung

◦ ◦ ◦ ◦

1

Giới thiệu

2

Lý thuyết

3

Các bước thực hiện

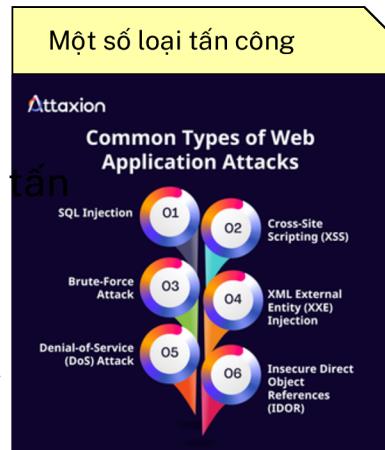
4

Kết quả và đánh giá

1. Giới thiệu

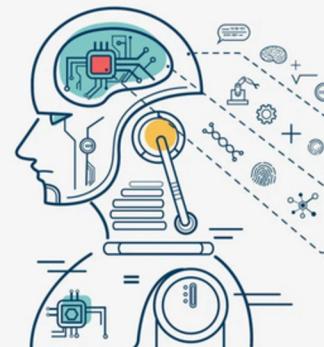


Trong bối cảnh số hóa ngày càng mạnh mẽ, việc phòng chống tấn công mạng vào ứng dụng web đang trở thành mối quan tâm hàng đầu trước các kỹ thuật tấn công phổ biến như SQL Injection, XSS, Path Traversal và OS Command Injection, nhằm khai thác các lỗ hổng bảo mật để xâm nhập hệ thống, đánh cắp dữ liệu nhạy cảm và gây rối loạn hoạt động.



Phân tích log web là một giải pháp đảm bảo an toàn chủ động, giúp doanh nghiệp phát hiện và ngăn chặn các cuộc tấn công một cách hiệu quả. Bằng cách sử dụng thuật toán học máy, ta có thể nhanh chóng xây dựng một hệ thống phòng thủ tự động vững chắc, tự động và có độ chính xác cao

Học máy



2. Lý thuyết



1

Các loại tấn công

SQL Injection



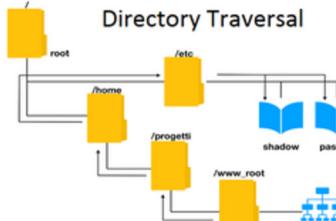
Kẻ tấn công **chèn mã SQL độc hại** vào đầu vào của ứng dụng web, từ đó cho phép thao túng cơ sở dữ liệu, dẫn đến rò rỉ dữ liệu, sửa đổi hoặc xóa thông tin quan trọng.

Cross-Site Scripting



Kẻ tấn công **chèn mã JavaScript độc hại** vào trang web. Khi người dùng truy cập, mã này được thực thi, có thể đánh cắp thông tin nhạy cảm như cookie hoặc tài khoản.

Path Traversal



Kẻ tấn công chỉnh sửa đầu vào và đường dẫn của ứng dụng web để **truy cập file ngoài phạm vi cho phép**, nhằm mục đích đọc, sửa hoặc xóa file nhạy cảm trên máy chủ.

OS Command Injection



Command Injection

Kẻ tấn công chèn và thực thi **lệnh hệ điều hành** thông qua đầu vào của ứng dụng, cho phép kiểm soát hoặc phá hoại hệ thống.

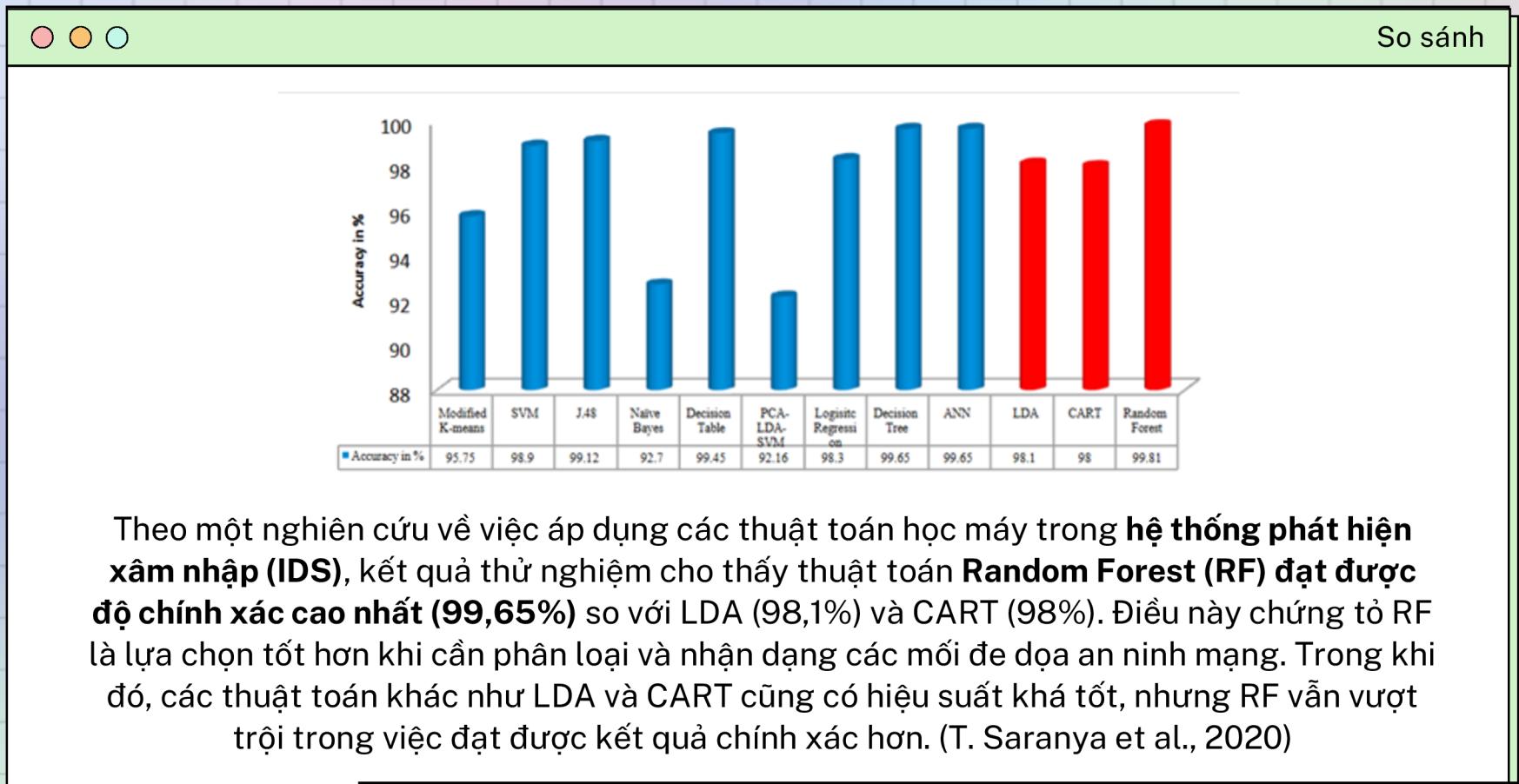
Điểm chung: lợi dụng lỗ hổng không kiểm tra đầu vào của ứng dụng web, chèn những lệnh hoặc chỉ dẫn không an toàn vào nhằm thao túng hệ thống

2. Lý thuyết

o o o o

2

Thuật toán Random Forest



T. Saranya, S. Sridevi, C. Deisy, T.D. Chung, M.K. Khan

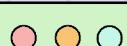
Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review, Procedia Computer Science, vol. 171, Elsevier B.V. (2020), pp. 1251-1260

2. Lý thuyết



2

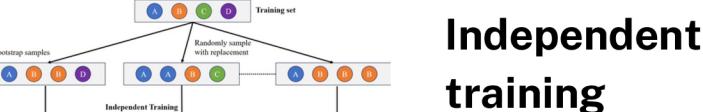
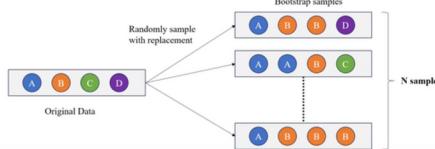
Thuật toán Random Forest



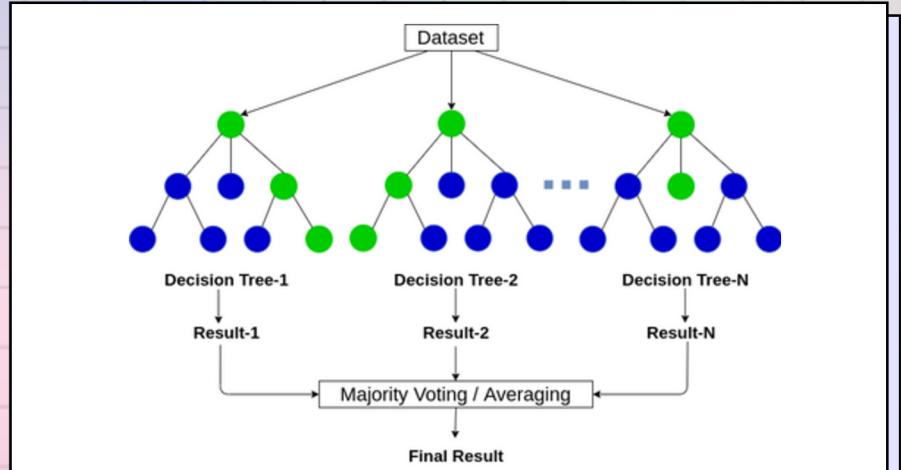
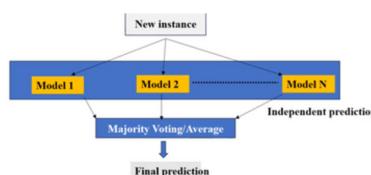
So sánh

- Bagging** (hay còn gọi là **Bootstrap Aggregation**) là một thuật toán **Supervised Ensemble Learning** (Kết hợp nhiều mô hình). Bagging gồm 3 bước cơ bản:

Boot-strapping



Aggregation



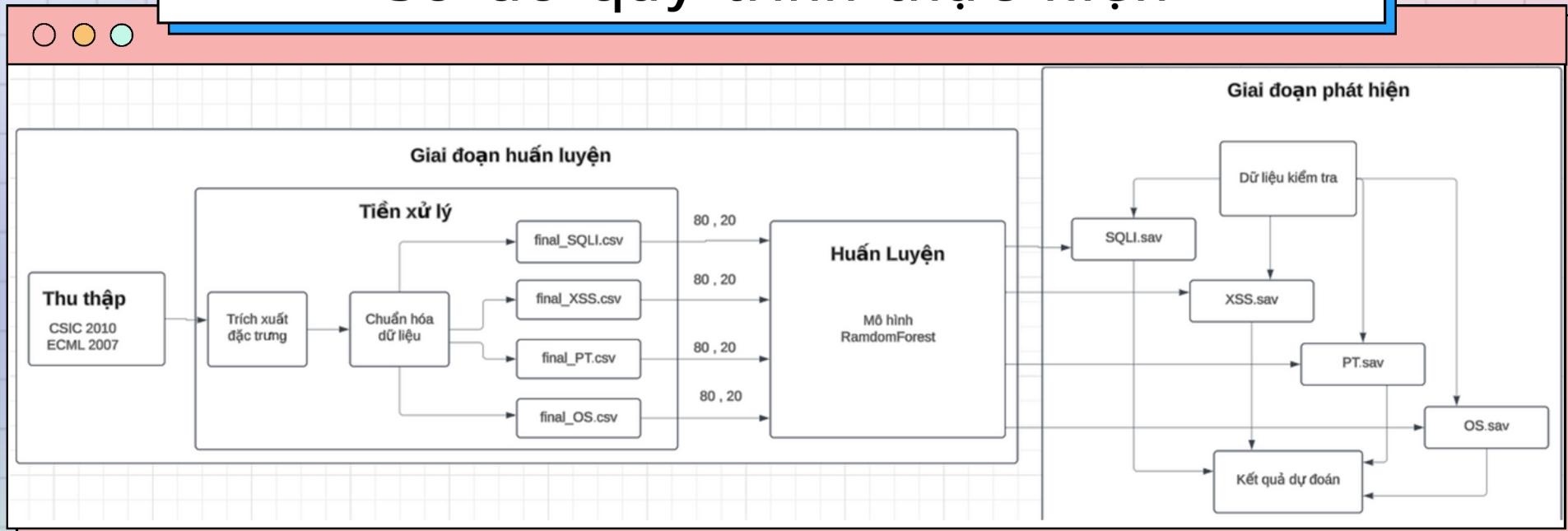
Random Forest là một cách áp dụng Bagging, với một số điểm khác biệt:

- Random Forest chỉ sử dụng mô hình cây quyết định (decision tree)
- Trong bước bootstrapping, chỉ **chọn ngẫu nhiên một tập nhỏ các đặc trưng (cột)** của tập dữ liệu thay vì chọn toàn bộ

3. Các bước thực hiện

...

Sơ đồ quy trình thực hiện



2

Tiền xử lý

Tiền xử lý gồm hai bước chính:

- Thực hiện chuẩn hóa URL đối với từng kiểu tấn công và gán nhãn **label tương ứng**
 - Unquote URL
 - Loại bỏ \n, ‘ ’, +
- **Trích xuất đặc trưng** của từng kiểu tấn công rồi chuyển dữ liệu về **dạng vector để tạo tập huấn luyện** với các chiều là các đặc trưng tương ứng

Unquote (Giải mã) các URL

Reserved characters after percent-encoding

%21	%23	%24	%26	%27	%28	%29	%2A	%2B
,	/	:	;	=	?	@	[]
%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

Trích xuất Đặc trưng và Tạo Vector

type	url
0	Valid /_L@/_feu1vhptpass2/1nieOnnnvnzktuassain/tg1ar...
1	Valid /inssgtz?eltdsstbw7/eQhmsdw7/imdb0tet/et/h...
2	Valid /fonRt/7N-D-4BR5xb@TU/qghew.cfm?
3	Valid /dyylkl.xD9cpu/4ot0ta/ts6xvrp1/hsh/a2cuerht/s...
4	Valid /2m6vlb1r37jSPC/cWVv/Mbar/oqr0/msc/etceebwgl/...
...	...
126424	Valid /antoanweb/publico/registro.jsp?modo=registro&...
126425	Valid /antoanweb/publico/registro.jsp?modo=registro&...
126426	Valid /antoanweb/publico/registro.jsp?modo=registro&...
126427	Valid /antoanweb/publico/registro.jsp?modo=registro&...
126428	Valid /antoanweb/publico/registro.jsp?modo=registro&...



type	url	label	"	xp	all	=	any	+	>	...	:	like	-uni
0	Valid /_L@/_feu1vhptpass2/1nieOnnnvnzktuassain/tg1ar...	0	0	0	1	0	1	0	...	0	0	1	
1	Valid /inssgtz?eltdsstbw7/eQhmsdw7/imdb0tet/et/h...	0	1	0	1	0	1	0	...	0	0	1	
2	Valid /fonRt/7N-D-4BR5xb@TU/qghew.cfm	0	0	0	0	0	0	0	...	0	0	1	
3	Valid /dyylkl.xD9cpu/4ot0ta/ts6xvrp1/hsh/a2cuerht/s...	0	0	0	1	0	1	0	...	0	0	0	
4	Valid /2m6vlb1r37jSPC/cWVv/Mbar/oqr0/msc/etceebwgl/...	0	0	0	1	0	1	0	...	0	0	0	
...	
125971	Valid /antoanweb/publico/registro.jsp?modo=registro&...	0	0	0	1	0	0	0	...	0	0	0	
125972	Valid /antoanweb/publico/registro.jsp?modo=registro&...	0	0	0	1	0	0	0	...	0	0	0	
125973	Valid /antoanweb/publico/registro.jsp?modo=registro&...	0	0	0	1	0	0	0	...	0	0	0	
125974	Valid /antoanweb/publico/registro.jsp?modo=registro&...	0	0	0	1	0	0	0	...	0	0	0	
125975	Valid /antoanweb/publico/registro.jsp?modo=registro&...	0	0	0	1	0	0	0	...	0	0	0	

2

Tiền xử lý

Các đặc trưng là các ký tự, từ khóa được sử dụng nhiều trong mỗi kiểu tấn công, được thu thập thủ công và tham khảo từ dự án tổng hợp payload mở [PayloadsAllTheThings](#)

SQL Injection



/**, %, +, ;, #, =, [], (), ^,
*, char, , -, <, >, .., |, ", <>,
<=, >=, &&, ||, :, !=, count,
into, or, and, not, null,
select, union, #, insert,
update, delete, drop,
replace, all, any, from,
count, user, where, sp, xp,
like, exec, admin, table,
sleep, commit, (),
between

=> 59 đặc trưng

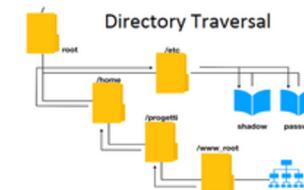
Cross-Site Scripting



&, %, /, \, +, ?, !, ;, #, =, [,
], \$, (), ^, *, , -, <, >, @, _, :,
, {, }., |, <>, ' , <>, [], ==, &#,
document, window,
iframe, location, this,
onload, onerror,
createElement,
String.fromCharCode,
search, div, img, <script,
src, href, cookie, var,
eval(), http, .js,

=> 53 đặc trưng

Path Traversal



./, ..\\, etc, passwd, \\., \\\/
./, /, :, //, :, /, system, ini, ..,
exec, :\%, %00, .bat, file,
windows, boot, winnt,
.conf, access, log

=> 26 đặc trưng

OS Command Injection



./, ..\\, etc, passwd, \\., \\\/
./, :, :, ., system32,
display, .exe, cmd, dir, :,
tmp/, etc/passwd, wget,
cat, ping, bash, ftp, |, ...,
exec, :\%, .bat, file, script,
rm, c:, winnt, access, log,
' , www., http, , bin/,
telnet, echo, root, -aux,
shell, uname, IP

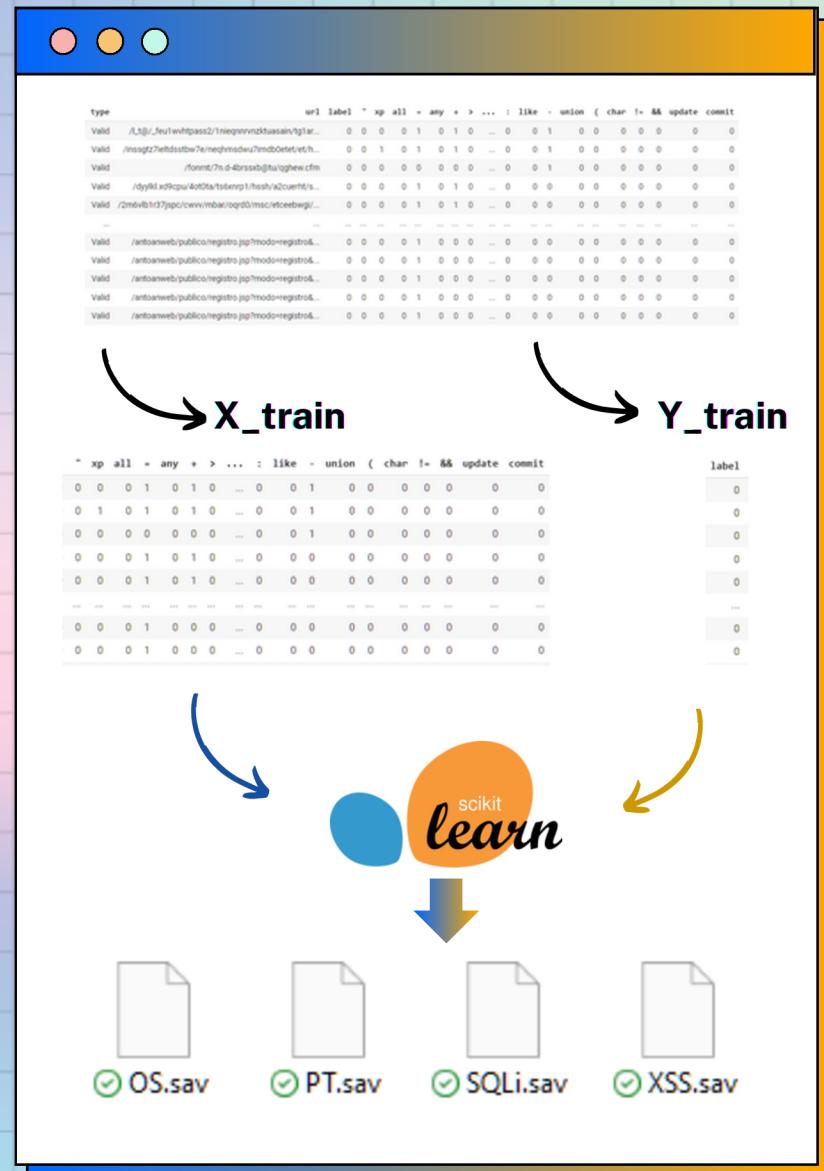
=> 46 đặc trưng

3

Huấn luyện mô hình

Đối với mỗi kiểu tấn công, tạo **một tập dữ liệu** ở bước 2 và **huấn luyện một mô hình** ở bước 3:

- Chia tập dữ liệu huấn luyện thành tập các biến huấn luyện **X_train** (gồm các thuộc tính) và tập các biến mục tiêu **Y_train** (nhãn)
- Tiến hành huấn luyện mô hình random forest sử dụng thư viện sklearn
- Kiểm tra kết quả bằng cách đưa tập thử nghiệm **X_test**, **Y_test** qua mô hình đã huấn luyện



3

Huấn luyện mô hình

Chuẩn bị các tập dữ liệu huấn luyện và thử nghiệm

```
# @title Chuẩn bị các tập dữ liệu huấn luyện và thử nghiệm

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

# Tạo một DataFrame X_df chỉ chứa các thuộc tính (Biến huấn luyện)
X = list(df.columns[3:])
X_df = df[X]

# Tạo một DataFrame Y_df chứa các label (Biến mục tiêu)
Y_df = df['label']

# Tách các tập huấn luyện thành tập thử nghiệm với kích thước 0.2
X_train, X_test, Y_train, Y_test = train_test_split(X_df, Y_df,
                                                    test_size=0.2, random_state=0)
```

```
# @title Huấn luyện mô hình và in ra kết quả

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
#Tạo model RandomForest RF và bắt đầu huấn luyện
RF = RandomForestClassifier(max_depth=19, n_estimators=124) # Use max_depth=19, n_estimators=124
RF.fit(X_train,Y_train)

print('Accuracy of Random Forest classifier on training set of SQLi: {:.2f}'.format(RF.score(X_train, Y_train)))
print('Accuracy of Random Forest classifier on test set of SQLi: {:.2f}'.format(RF.score(X_test, Y_test)))

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

# Create the confusion matrix
# Changed y_test to Y_test to match the variable name used in train_test_split
cm = confusion_matrix(Y_test, Y_pred)

ConfusionMatrixDisplay(confusion_matrix=cm).plot();
metrics.accuracy_score(Y_test, Y_pred)

accuracy = accuracy_score(Y_test, Y_pred)
precision = precision_score(Y_test, Y_pred) # Use precision_score to calculate precision
recall = recall_score(Y_test, Y_pred) # Use recall_score to calculate recall
F1_score = f1_score(Y_test,Y_pred) # Use f1_score to calculate F1 score
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1_score",F1_score);
```

Ví dụ với huấn luyện mô hình phát hiện SQLi

4. Kết quả huấn luyện



Accuracy

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

- Accuracy là tỷ lệ số lượng URI được phân loại đúng (gồm cả URI bình thường và URI tấn công) trên tổng số URI.

Precision

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

- 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).

Recall

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

- 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).

F1-Score

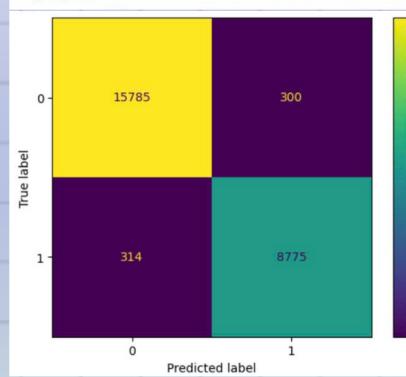
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- F1-score là trung bình điều hòa giữa Precision và Recall, một sự kết hợp 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.

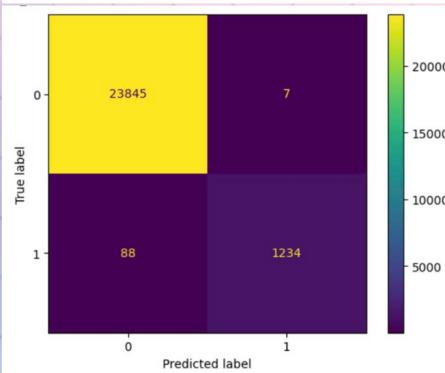
4. Kết quả huấn luyện



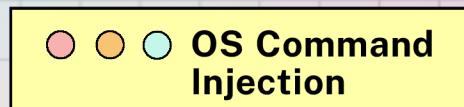
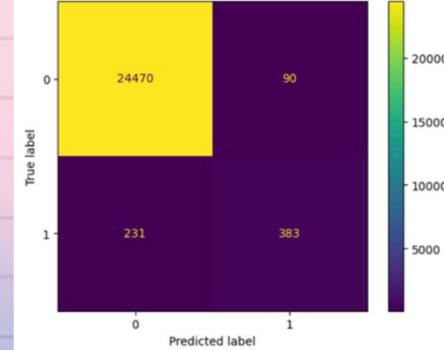
Accuracy: 0.975609756097561
Precision: 0.9669421487603306
Recall: 0.9654527450764661
F1_score 0.9661968729354767



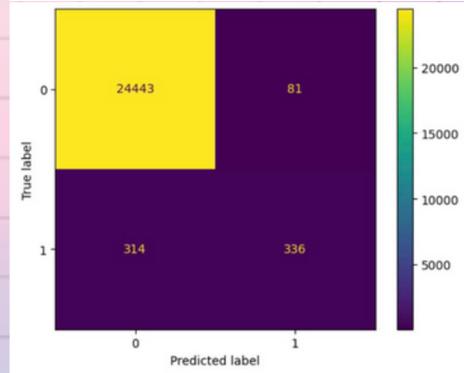
Accuracy: 0.9962262651942481
Precision: 0.9943593875906527
Recall: 0.9334341906202723
F1_score 0.962934061646508



Accuracy: 0.9872487487089855
Precision: 0.8097251585623678
Recall: 0.6237785016286646
F1_score 0.7046918123275069



Accuracy: 0.9843092079129261
Precision: 0.8057553956834532
Recall: 0.5169230769230769
F1_score 0.6298031865042174



Kết quả: Kết quả cho thấy mô hình hoạt động tốt nhất trong phát hiện SQL Injection và Cross-Site Scripting (XSS) với độ chính xác lần lượt là 97.56% và 99.63%, cùng các chỉ số Precision, Recall, và F1-Score đều cao, chứng tỏ khả năng phân loại chính xác và ít bỏ sót. Tuy nhiên, đối với Path Traversal và OS Command Injection, dù độ chính xác đạt trên 98%, các chỉ số Recall thấp (lần lượt là 62.37% và 51.69%) cho thấy mô hình còn bỏ sót nhiều tấn công. Cần cải thiện khả năng phát hiện các trường hợp này để tăng hiệu quả bảo mật toàn diện.

o o o o

Cảm ơn thầy và các
bạn đã lắng nghe

SQL Injection
XSS
Path Traversal
OS Command