ỨNG DỤNG XỬ LÝ DỮ LIỆU LỚN PHÁT HIỆN TẤN CÔNG ỨNG DỤNG WEB THEO THỜI GIAN THỰC

Nguyễn Huỳnh Quốc Doanh [CH2020203]
Võ Chánh Đại [CH2020202]
Võ Xuân Khang [CH2002008]

Nhóm 24
Môn học: Xử lý dữ liệu lớn - IT2034.CH1502
Phòng Đào tạo Sau Đại Học và Khoa học Công nghệ
Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh

Tóm tắt. Trong đồ án môn học Xử lý dữ liệu lớn này, nhóm học viên thực hành triển khai một hệ thống ứng dụng xử lý dữ liệu lớn phát hiện tấn công web theo thời gian thực ứng dụng học máy. Hệ thống này được nghiên cứu nhằm hỗ trợ cho việc hiện đại hoá hệ thống quản lý log và sự kiện tập trung (SIEM), vốn được xây dựng để thu thập và xử lý các các sự kiện bảo mật tại một điểm, trong điều kiện SIEM ngày càng xuất hiện nhiều điểm yếu về lưu trữ và độ chính xác của các cảnh báo tấn công.

Từ khoá: xử lý dữ liệu lớn, thời gian thực, phát hiện bất thường, tấn công web, học máy, học sâu.

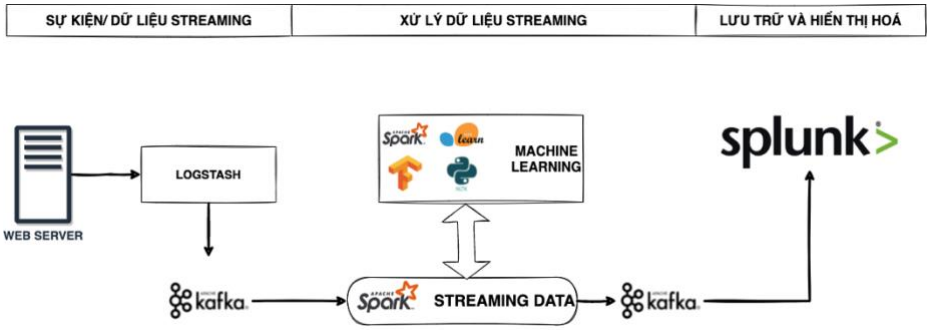
1 Giới thiệu

Hệ thống quản lý log và sự kiện tập trung (SIEM) được xây dựng để thu thập và xử lý các các sự kiện bảo mật tại một điểm. Hiện nay SIEM tồn tại hai điểm yếu phổ biến: thứ nhất, SIEM tạo ra lượng lớn các cảnh báo giả ở các hệ thống phát hiện tấn công chủ yếu dựa vào tập các "chữ ký" đã biết; thứ hai, với lượng dữ liệu ngày càng tăng, việc xử lý sự kiện và lưu trữ tập trung dễ dàng xuất hiện các vấn đề về hiệu suất vận hành và chi phí cho lưu trữ log thô. Trong nghiên cứu này, chúng tôi thực hành triển khai một hệ thống ứng dụng xử lý dữ liệu lớn phát hiện tấn công ứng dụng web theo thời gian thực để chứng minh khả năng khắc phục một phần điểm yếu của hai vấn đề trên.

Bài trình bày bao gồm phương pháp đề xuất ở phần 2 và kết quả nhóm đã đạt được trong phần 3. Phần 4 nhóm đề xuất các hướng phát triển tiếp theo của chủ đề này và đưa ra kết luận về bài nghiên cứu.

2 Phương pháp

2.1 Mô hình hệ thống xử lý dữ liệu lớn



Hình 1 - Minh hoạ quy trình xử lý dữ liệu lớn

Hệ thống xử lý dữ liệu lớn bao gồm các giai đoạn và các công nghệ hỗ trợ như sau
Thứ nhất, giai đoạn sinh ra dữ liệu

- Máy chủ web Apache: đang vận hành ứng dụng web DVWA, đây là ứng dụng với mục tiêu chính là hỗ trợ các chuyên gia bảo mật kiểm tra kỹ năng và công cụ của họ trong môi trường hợp pháp, giúp các nhà phát triển web hiểu rõ hơn về các quy trình bảo mật ứng dụng web và hỗ trợ giáo viên / sinh viên dạy / học bảo mật ứng dụng web trong môi trường lớp học.
- Logstash thực hiện chuyển tiếp thời gian thực log sinh ra từ máy chủ web vào Kafka.
- Kafka: xây dựng một "kafka topic" xử lý dòng dữ liệu trong thời gian thực nhận từ logstash.

Thứ hai, xử lý dòng dữ liệu

- Thành phần Học máy: thực hiện huấn luyện, kiểm tra và lưu trữ các mô hình học máy. Các mô hình này được lưu lại dùng cho quá trình dự đoán tấn công web thời gian thực trên dòng dữ liệu.
- Spark Streaming: đọc dữ liệu dòng từ Kafka, tiên đoán tấn công dữ liệu web nhờ mô hình đã được huấn luyện trước đó và chuyển tiếp log kèm kết quả đến Splunk.

Thứ ba, lưu trữ và hiển thị hoá

- Splunk Security: nhận dữ liệu sau khi đã xử lý và tiến hành lưu trữ, đồng thời phục vụ tạo các "dashboard" theo dõi về kết quả dự đoán tấn công từ quá trình xử lý dữ liệu lớn và học máy. Từ đó, cung cấp cơ sở cho người phân tích bảo mật ra quyết định.

2.2 Chuẩn bị dữ liệu

Các dữ liệu liên quan dự án này nhắc đến dữ liệu dành cho học máy và dữ liệu phục vụ minh hoạ ứng dụng

Thu thập dữ liệu

Nhóm đã thực hiện thu thập dữ liệu tấn công web bằng cách dựng một ứng dụng web và dùng công cụ "sqlmap" tiến hành kiểm thử nó. Những thành phần truyền vào (parameter) tại các điểm nhập (input) của ứng dụng web sau đó được thu thập và đánh nhãn như những sự kiện "bất thường (Hình 2)

Bên cạnh đó, một tập dữ liệu truy cập web bình thường được thu thập từ tập dữ liệu công khai tham khảo từ tài liệu 2 (Hình 3)

sqli.csv

```
Sentence,Label
a,1
a' ,1
a' --,1
a' or 1 = 1; --,1
@,1
?,1
' and 1 = 0 )  union all,1
? or 1 = 1 --,1
x' and userid is NULL; --,1
x' and email is NULL; --,1
anything' or 'x' = 'x,1
x' and 1 =  ( select count ( * )  from tabname ) ; --,1
x' and members.email is NULL; --,1
x' or full_name like '%bob%,1
23 or 1 = 1; --,1
'; exec master..xp_cmdshell 'ping 172.10.1.255'--,1
a,1
1 or 1 = 1,1
1' or '1' = '1,1
1 and user_name (  )   =  'dbo',1
1,1
1'1,1
1 exec sp_  ( or exec xp_ ) ,1
1 and 1 = 1,1
1' and 1 =  ( select count ( * )  from tablenames ) ; --,1
```

Hình 2 - Mẫu tập dữ liệu đánh nhãn 1 (tấn công SQL injection)

sqli.csv

```
" I stand humanity, though I would make kind, I would make true",0
" Let us affront reprimand smooth mediocrity squalid contentment times, hurl face custom, trade, office, fact upshot history, great r
" Where is, nature",0
" He measures you, men, events",0
" Ordinarily, every body society reminds us somewhat else, person",0
" Character, reality, reminds nothing else; takes place whole creation",0
" The man must much, must make circumstances indifferent",0
" Every true man cause, country, age; requires infinite spaces numbers time fully accomplish design; — posterity seem follow steps tr
" A man Caesar born, ages Roman Empire",0
" Christ born, millions minds grow cleave genius, confounded virtue possible man",0
" An institution lengthened shadow one man; as, Monachism, Hermit Antony; Reformation, Luther; Quakerism, Fox; Methodism, Wesley; Abo
" Scipio, Milton called height Rome; history resolves easily biography stout earnest persons",0
 Standard automatic speech recognizers output unstructured streams words,0
" They neither perform proper segmentation output sentences, predict punctuation symbols",0
" The unavailable punctuations sentence boundaries transcribed speech texts create barriers many subsequent processing tasks, summari
" Thus, segmentation long texts necessary many real applications",0
" For example, speech-to-speech translation, continuously transcribed speech texts need segmented fed subsequent machine translation
", 1998; Nakamura, 2009)",0
" This current machine translation (MT) systems perform translation sentence level, various models used MT trained segmented sentence
 The punctuation prediction problem attracted research interest speech processing community natural language processing community,0
" Most previous work primarily exploits local features statistical models lexicons, prosodic cues hidden event language model (HELM)
", 2005; Matusov et al",0
", 2006; Huang Zweig, 2002; Stolcke Shriberg, 1996)",0
 The word-level models integrating local features narrow views input could achieve satisfied performance due limited context informat
", 2008)",0
```

Hình 3 - Mẫu tập dữ liệu đánh nhãn 0 (bình thường)

Trích xuất dữ liệu

```
# Vectorization
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
print(" Vectorization ---------------- Start")
vectorizer = CountVectorizer(min_df=2, max_df=0.7, max_features=4096, stop_words=stopwords.words('english'))
posts = vectorizer.fit_transform(df['Sentence'].values.astype('U')).toarray()
posts.shape = (4200, 64, 64, 1)
X = posts
y = df['Label']
print(" Vectorization ---------------- Done")
```

Hình 4 - Vector hoá dữ liệu

Tập dữ liệu sau khi được vector hoá nhờ thư viện "sklearn" được chia thành tập huấn luyện (chiếm 80%) và tập kiểm tra (20%). Hình 4 thể hiện rằng dữ liệu thô được đưa vào tiến hành trích xuất thuộc tính với lượng thuộc tính tối đa 4096 và loại bỏ các "stop word".

2.3 Thuật toán

Các mô hình thuật toán lần lượt được áp dụng sử dụng các thư viện sklearn, tensorflow bao gồm: Naive Bayes, SVM, KNN, Decision Tree, CNN. Bảng 1 so sánh các chỉ số đáng giá các thuật toán sau quá trình kiểm tra (test).

Bảng 1 - So sánh các thuật toán

Tiêu chí	Naive Bayes	SVM	KNN	Decision Tree	CNN
Độ chính xác (Accuracy)	0.9774	0.7643	0.5976	0.8667	0.9690
"Precision"	0.9299	1.0	0.4271	0.6923	0.9185
"Recall"	1.0	0.2143	1.0	1.0	0.9841

Các phương pháp sử dụng Naive Bayes và CNN cho độ chính xác cao, nhóm quyết định đưa mô hình đã lưu lại của Naive Bayes và CNN vào dự đoán trên dòng dữ liệu. Tuy nhiên với CNN không thể hoạt động được do mô hình lưu lại quá lớn không phù hợp. Trong khi đó Naive Bayes là có thể hoạt động tốt với dòng dữ liệu. Mô hình được lưu lại dùng "pickle".

```
# Naive Bayes

from sklearn.naive_bayes import GaussianNB

print(" Naive Bayes ---------------- Start")
gnb = GaussianNB()
gnb.fit(trainX, y_train)
pred_gnb = gnb.predict(testX)
print(" Naive Bayes ---------------- Done")
```

Hình 5 - Minh hoạ code huấn luyện và kiểm tra sử dụng Naive Bayes

2.4 Dự đoán tấn công SQL Injection trên dòng dữ liệu

Mô hình học máy với Naive Bayes được đưa vào dự đoán với việc xây dựng một UDF trong Spark. Dữ liệu được truyền thời gian thực từ Kafka được Spark Stream xử lý, sau đó gọi hàm dự đoán và chuyển tiếp đến Splunk (trong trường hợp này Splunk đọc giám sát trực tiếp thư mục đầu ra) (Hình 7)

```
myvectorizer = pickle.load(open("vectorizer", 'rb'))
f = open('sqli_classifier.pickle','rb')
mymodel = pickle.load(f)

def predict_sqli_attack(input_val):
    input_val=clean_data(input_val)
    input_val=[input_val]
    input_val = myvectorizer.transform(input_val).toarray()
    input_val.shape=(1,4096)
    result=mymodel.predict(input_val)
    return str(result)
convertUDF_predict = udf(lambda z: predict_sqli_attack(z))
```

Hình 6 - Spark UDF với chức năng dự đoán

```
df = spark\
  .readStream\
  .format("kafka")\
  .option("kafka.bootstrap.servers", "localhost:9092")\
  .option("subscribe", "input-events")\
  .load()
df = df.selectExpr("CAST(value AS STRING)")
## Extract
df = df.select("value")
""" Converting function to UDF """
df = df.select(col("value"), \
    convertUDF_predict(col("value")).alias("predicted") )

query = df.writeStream\
        .format("csv")\
        .option("path", "./out_log")\
        .option("checkpointLocation","./checkpoint_out_log")\
        .start()
query.awaitTermination()
```

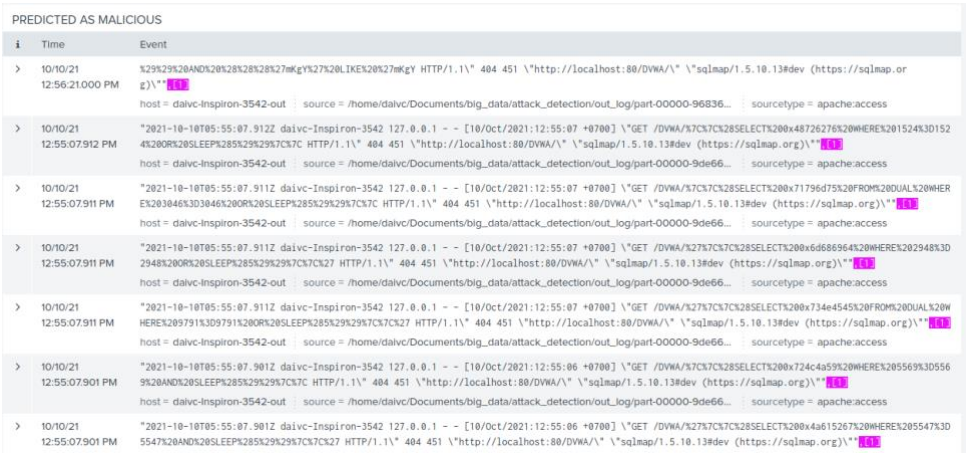
Hình 7 - Lấy dữ liệu vào, dự đoán và ghi kết quả

3 Kết quả

3.1 Xử lý log trước khi đưa vào SIEM

Như vậy dữ liệu đã được "streaming" qua Spark Streaming, xử lý trước khi được đưa vào index tại Splunk.

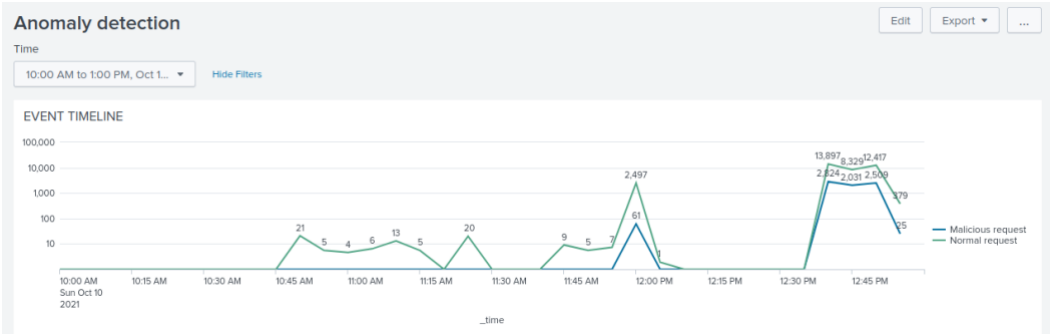
Kết quả dự đoán bất thường (đánh dấu [1]) và bình thường được thể hiện ngay trên sự kiện đảm bảo sự kiện log được lưu trữ đầy đủ từ nguồn cho đến Splunk.



Hình 8 - Sự kiện được học máy dự đoán là "tấn công

3.2 Phát hiện tấn công dựa trên bất thường

Các kết quả dự đoán tấn công được "gắn" vào sự kiện log và chuyển tiếp lưu trữ trên Splunk. Từ kết quả đó, nhóm học viên đã vẽ được biểu đồ thể hiện lượng sự kiện được đánh giá là bình thường và bất thường theo thời gian thực. Biểu đồ 1 cung cấp cái nhìn trực quan cho người phân tích bảo mật về lưu lượng tấn công.



Biểu đồ 1 - Lượng truy cập web bất thường (xanh dương) và bình thường theo thời gian

4 Kết luận và hướng phát triển

Như vậy trong phạm vi đề tài, nhóm học viên đã thể hiện kết quả học được từ môn Xử lý dữ liệu lớn với việc xây dựng hoàn thành một hệ thống xử lý dữ liệu lớn phục vụ một bài toán cụ thể trong lĩnh vực an toàn thông tin. Trong hệ thống đó, dữ liệu thay vì được giám sát và chuyển trực tiếp vào hệ thống tập trung log (SIEM) thì đã được xử lý trước nhờ sự hỗ trợ của công cụ xử lý dữ liệu lớn, cụ thể là Spark Streaming. Bên cạnh đó, bước đầu các thuật toán học máy và học sâu đã được áp dụng để dự đoán hành vi

tấn công dựa trên phân tích log, điều này bổ sung năng lực cho hệ thống phát hiện tấn công và cung cấp cơ sở cho việc ra quyết định của người phân tích bảo mật.

Với ý tưởng và mô hình được xây dựng một cách cơ bản, bài nghiên cứu này được xem là phiên bản khởi tạo của quá trình chuyển đổi ứng dụng xử lý dữ liệu lớn vào hiện đại hoá SIEM. Trong đó, việc xử lý dữ liệu có thể nâng cấp tiếp tục bằng cách cải tiến các mô hình học máy và học sâu, ứng dụng PCA và các thành phần hỗ trợ khác trong giảm chiều dữ liệu và tiến hành kiểm tra hiệu năng trên đa dạng mô hình. Trong nghiên cứu này nhóm cũng chỉ dừng lại ở việc dự đoán tấn công "SQL injection", hướng đi này hoàn toàn có thể mở rộng với những loại tấn công web phổ biến khác như Từ chối dịch vụ,…. Cuối cùng, ở phần khai thác kết quả dự đoán người quản trị có thể thiết đặt thêm cảnh báo tự động dựa trên kết quả sinh ra từ mô hình dự đoán.

Tài liệu tham khảo

1. Habeeb, Riyaz Ahamed Ariyaluran, et al. "Real-time big data processing for anomaly detection: A survey." International Journal of Information Management 45 (2019): 289-307.
2. Tekerek, Adem. "A novel architecture for web-based attack detection using convolutional neural network." Computers & Security 100 (2021): 102096.
3. Riera, Tomás Sureda, et al. "Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review." Sustainability 12.12 (2020): 1-45.