

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN

-----oOo-----



KHÓA LUẬN TỐT NGHIỆP

**ĐỀ TÀI: NGHIÊN CỨU CÁC KỸ THUẬT PHÁT HIỆN LƯU
LƯỢNG BẤT THƯỜNG TRÊN HỆ THỐNG MẠNG**

Giảng viên hướng dẫn: ThS. Trần Đức Tốt.

Sinh viên thực hiện:

1. Nguyễn Bảo Kha - 2033180168.
2. Đoàn Thị Thu Hà - 2033181104.
3. Nguyễn Bá Khánh Duy - 2033101110.

TP. HCM, ngày ... tháng ... năm 2022

LỜI CẢM ƠN

Được sự phân công của Khoa Công nghệ thông tin - Trường Đại học Công nghiệp Thực phẩm TP.HCM và sự đồng ý của giảng viên hướng dẫn ThS. Trần Đắc Tốt, nhóm em đã thực hiện đề tài “Nghiên cứu các kỹ thuật phát hiện lưu lượng bất thường trên hệ thống mạng”.

Nhóm em xin chân thành cảm ơn giảng viên hướng dẫn cùng toàn thể Quý thầy/cô giảng viên - Trường Đại học Công nghiệp Thực phẩm đã dạy cho chúng em những kiến thức quý báu trong suốt thời gian học tập tại trường để nhóm có thể hoàn thành khóa luận tốt nghiệp.

Do kiến thức và khả năng lý luận của nhóm còn nhiều hạn chế nên khóa luận vẫn còn những thiếu sót nhất định. Nhóm rất mong nhận được những lời góp ý từ các thầy/cô giảng viên để khóa luận tốt nghiệp của nhóm được hoàn thiện tốt hơn.

Sau cùng, nhóm xin kính chúc Quý thầy/cô Ban lãnh đạo và các phòng ban chức năng Trường Đại học Công nghiệp Thực phẩm TP.HCM dồi dào sức khỏe và thành công trong sự nghiệp.

Nhóm xin chân thành cảm ơn!

LỜI CAM KẾT

Nhóm em xin cam đoan khóa luận tốt nghiệp về đề tài “Nghiên cứu các kỹ thuật phát hiện lưu lượng bất thường trên hệ thống mạng” là trung thực và chưa hề được sử dụng để bảo vệ ở bất kỳ luận văn nào. Mọi thông tin trích dẫn trong khóa luận đã được ghi rõ nguồn gốc rõ ràng và được phép công bố.

Hồ Chí Minh, ngày... tháng 08 năm 2022.

Sinh viên thực hiện

(Ký, ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Nhóm sinh viên gồm: 1. Nguyễn Bảo Kha MSSV: 2033180168.
 2. Đoàn Thị Thu Hà MSSV: 2033181104.
 3. Nguyễn Bá Khánh Duy MSSV: 2033181110.

Nhận xét:

.....

.....

.....

.....

.....

.....

.....

.....

Điểm đánh giá:

.....

.....

.....

Ngày ... Tháng ... Năm 2022.

(Ký, ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	i
LỜI CAM KẾT	ii
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	iii
MỤC LỤC	iv
PHẦN MỞ ĐẦU	1
CHƯƠNG 1: TÌM HIỂU VỀ HỆ THỐNG GIÁM SÁT AN NINH SỰ KIỆN	2
1.1. Tổng quan về hệ thống giám sát an ninh sự kiện.....	2
1.1.1. Tổng quan	2
1.1.2. Tầm quan trọng của SIEM.....	2
1.1.3. Các thành phần chính của SIEM.....	3
1.1.4. Lợi ích của SIEM	4
1.2. Một số giải pháp triển khai SIEM	4
1.2.1. Giải pháp McAfee SIEM	4
1.2.1.1. Giới thiệu	4
1.2.1.2. Các giải pháp SIEM của McAfee.....	4
1.2.1.3. Các thành phần chính của McAfee SIEM	5
1.2.2. Giải pháp IBM Qradar SIEM.....	5
1.2.2.1. Tổng quan.....	5
1.2.2.2. Các dữ liệu được thu thập và phân tích bởi Qrada SIEM.....	6
1.2.2.3. Lợi ích của IBM Qrada SIEM.....	6
1.2.3. Giải pháp bảo mật Splunk SIEM	7
1.2.3.1. Tổng quan về giải pháp Splunk SIEM.....	7
1.2.3.2. Các thành phần kiến trúc của Splunk	7
1.2.3.3. Lợi ích của giải pháp Splunk SIEM	8

1.2.3.4. Các sản phẩm Splunk SIEM cung cấp.....	9
1.3. Nghiên cứu các bộ dataset phục vụ cho dự đoán bất thường HDFS, HADOOP	9
1.3.1. Tìm hiểu HADOOP	9
1.3.1.1. Tổng quan về Hadoop.....	9
1.3.1.2. Cấu trúc Hadoop.....	10
1.3.1.2.1. Hadoop Distributed File System (HDFS):	10
1.3.1.2.2. Hadoop MapReduce:.....	10
1.3.1.2.3. Hadoop Common:.....	10
1.3.1.2.4. Hadoop YARN:	11
1.3.1.3. Hadoop hoạt động như thế nào?.....	11
1.3.1.4. Ưu điểm Hadoop:	11
1.3.2. Tìm hiểu về HDFS	12
1.3.2.1. Tổng quan về HDFS:	12
1.3.2.2. Kiến trúc của HDFS:	12
1.3.2.3. Ưu điểm của HDFS:	12
1.3.2.4. Đặc trưng của HDFS:	12
1.4. Tìm hiểu một số hình thức tấn công trong Top 10 OWASP	13
1.4.1. SQJ Injection.....	13
1.4.1.1. SQL Injection là gì?.....	13
1.4.1.2. Cách thức hoạt động	14
1.4.1.3. Tác hại.....	14
1.4.1.4. Cách phòng chống SQL Injection	14
1.4.2. XSS (Cross Site Scripting)	14
1.4.2.1 XSS (Cross Site Scripting) là gì ?	14
1.4.2.2 Cách thức hoạt động	15

1.4.2.3	Tác hại.....	15
1.4.2.4	Cách phòng chống XSS	15
CHƯƠNG 2: TÌM HIỂU CÁC THUẬT TOÁN MÁY HỌC ỨNG DỤNG TRONG PHÁT HIỆN BẤT THƯỜNG		16
2.1.	Support Vector Machine (SVM).....	16
2.1.1.	Tổng quan SVM.....	16
2.1.2.	Ý tưởng của SVM	16
2.1.3.	Ưu điểm của SVM.....	18
2.1.4.	Nhược điểm của SVM.....	18
2.2.	Long Short Term Memory (LSTM).....	19
2.2.1.	Tổng quan LSTM.....	19
2.2.2.	Ý tưởng LSTM.....	20
2.2.3.	Thứ tự các bước của LSTM.....	21
2.2.4.	Kết luận.....	23
2.3.	Recurrent Neural Network (RNN)	24
2.3.1.	Tổng quan RNN	24
2.3.2.	Phương thức hoạt động RNN.....	24
2.3.3.	Ứng dụng của RNN.....	25
2.3.4.	Phân loại RNN	25
2.3.5.	Hạn chế của RNN.....	26
2.4.	Convolutional Neural Network (CNNs).....	27
2.4.1.	Tổng quan CNNs.....	27
2.4.2.	Điều gì khiến CNN trở nên hữu ích như vậy?.....	27
2.4.3.	Ứng dụng của CNN.....	27
2.4.4.	Tìm hiểu tính năng, lớp và phân loại	28
2.4.5.	Thực hiện thử nghiệm thuật toán	29

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG PHÁT HIỆN TẤN CÔNG MẠNG DỰA TRÊN LƯU LƯỢNG BẤT THƯỜNG TRONG HỆ THỐNG MẠNG	39
3.1 Mô hình triển khai	39
3.1.1. Thông tin thiết bị triển khai	39
3.2. Thực nghiệm kịch bản tấn công	40
3.2.1. Tấn công SQL Injection vào Web Server_Splunk.....	40
3.3.1.1 Mục đích	40
3.3.1.2 Quá trình tấn công	41
3.3.1.3 Kết quả.....	41
3.3.1.4 Đánh giá kịch bản.....	42
3.3.2 Tấn công XSS vào Web Server	42
3.3.2.1 Mục đích	42
3.3.2.2 Quá trình tấn công	42
3.3.2.3 Kết quả.....	43
3.3.2.4 Đánh giá kịch bản.....	43
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	44
TÀI LIỆU THAM KHẢO	45
LINK THAM KHẢO	45

DANH MỤC HÌNH ẢNH

Hình 1.1 Giải pháp SIEM	2
Hình 1.2 Các thành phần kiến trúc của Splunk SIEM	7
Hình 2.1 Đồ thị miêu tả bài toán phân 2 lớp SVM	17
Hình 2.2 Mô tả lề trong đồ thị của bài toán phân 2 lớp SVM	17
Hình 2.3 Sự lặp lại kiến trúc module trong mạng RNN chứa 1 tầng ẩn	19
Hình 2.4 Sự lặp lại kiến trúc module trong mạng LTSM chứa 4 tầng ẩn	20
Hình 2.5 Ký hiệu biểu diễn kiến trúc	20
Hình 2.6 Đường đi của ô trạng thái (Cell state) trong mạng LSTM	20
Hình 2.7 Một cổng của hàm sigmoid trong LSTM	21
Hình 2.8 Tầng cổng quên (Forget gate layer)	22
Hình 2.9 Cập nhật giá trị cho ô trạng thái bằng cách kết hợp 2 kết quả từ tầng cổng vào và tầng ẩn hàm tanh	22
Hình 2.10 Ô trạng thái mới	23
Hình 2.11 Điều chỉnh thông tin ở đầu ra thông qua hàm tanh	23
Hình 2.12 Hoạt động của RNN	24
DANH MỤC BẢNG BIỂU	
Bảng 4.1 Thông tin thiết bị triển khai.	40

PHẦN MỞ ĐẦU

Trong thời đại công nghệ phát triển như hiện nay, vấn đề đảm bảo sự an toàn cho hệ thống an ninh mạng doanh nghiệp luôn được đặt lên hàng đầu. Sự ra đời của các hệ thống giám sát an toàn mạng là một bước phát triển tiến bộ trong lĩnh vực an toàn thông tin mạng, tuy nhiên vẫn còn gặp nhiều khó khăn thách thức trong quá trình hoạt động. Bởi cùng với sự phát triển nhanh chóng của công nghệ, mà các kỹ thuật tấn công phức tạp cũng được “nâng cấp” trở nên đặc biệt hơn, tinh vi hơn để lẫn tránh khỏi sự phát hiện của các hệ thống bảo mật. Quản trị viên thường chỉ phát hiện ra tấn công khi đã có những thiệt hại nhất định trên hệ thống. Ngoài tấn công mạng, các hiểm họa tấn công từ chính trong mạng nội bộ, mạng LAN của cơ quan, tổ chức cũng là một trong những mối đe dọa an toàn thông tin nghiêm trọng. Các cuộc tấn công này rất khó bị phát hiện theo các cách thức và kỹ thuật thông thường. Bên cạnh đó, việc đào tạo, xây dựng đội ngũ nguồn nhân lực thực hiện giám sát mạng còn chưa được quan tâm đúng mức. Các chuyên gia giám sát an toàn mạng chưa phủ rộng được tất cả các cơ quan, tổ chức có sử dụng mạng hiện nay. Việc trang bị các kỹ năng thực hành cho đội ngũ quản trị viên để có được hiệu quả tốt trong giám sát an toàn mạng và ứng cứu sự cố mạng là một vấn đề khó khăn.

Hơn thế nữa, Chi phí cần thiết để xây dựng và duy trì một hệ thống giám sát an toàn mạng không nhỏ và không phải tổ chức nào cũng có thể đáp ứng được. Trong đó bao gồm chi phí phần cứng cần thiết để thu thập và phân tích lượng dữ liệu lớn được tạo ra từ các chức năng giám sát mạng, chi phí chi trả cho lực lượng chuyên gia để thực hiện phân tích giám sát an toàn mạng và chi phí để đầu tư cơ sở hạ tầng.

Thông thường hệ thống CNTT trong doanh nghiệp được quản lý qua nhiều bộ phận như: mạng, ứng dụng, phần mềm,... Do đó, khi có sự cố xảy ra việc tổng hợp nhật ký và sự kiện trong thời điểm đó là rất khó. Quá trình điều tra nguyên do bị tấn công, nguồn tấn công sau đó tiêu tốn rất nhiều thời gian, công sức nhưng lại không đảm bảo hiệu quả. Vì vậy, do nhiều nguyên nhân trên nên sự ra đời của các hệ thống giám sát an ninh mạng là vô cùng cần thiết. Với các tính năng, lợi ích cũng như chi phí hợp lý của hệ thống giám sát an ninh sự kiện (SIEM) sẽ là một lựa chọn thích hợp cho các doanh nghiệp hiện nay.

CHƯƠNG 1: TÌM HIỂU VỀ HỆ THỐNG GIÁM SÁT AN NINH SỰ KIỆN

1.1. Tổng quan về hệ thống giám sát an ninh sự kiện

1.1.1. Tổng quan



Hình 1.1 Giải pháp SIEM

Hệ thống giám sát an ninh sự kiện hay Security Information and Event Management (SIEM) là hệ thống kết hợp giữa hệ thống quản lý thông tin bảo mật (SIM) và quản lý sự kiện bảo mật (SEM) để cung cấp phân tích thời gian thực về các cảnh báo bảo mật do các ứng dụng và phần cứng mạng tạo ra.

SIEM tăng cường khả năng phát hiện các mối đe dọa, tuân thủ và quản lý sự cố bảo mật, thông qua việc thu thập và phân tích các nguồn, dữ liệu sự kiện bảo mật lịch sử và thời gian thực.

SIEM cũng cung cấp tính năng tổng hợp dữ liệu trên toàn bộ mạng doanh nghiệp và chuẩn hóa dữ liệu đó để phân tích thêm. Ngoài ra, SIEM giúp kích hoạt tính năng giám sát bảo mật, giám sát hoạt động bảo mật, giám sát hoạt động của người dùng và tuân thủ.[1]

1.1.2. Tầm quan trọng của SIEM

Khi hệ thống công nghệ thông tin của các doanh nghiệp được trang bị nhiều hãng và thiết bị công nghệ khác nhau như: Router, Switch, Server, Cơ sở dữ liệu, SAN,... Và

các thiết bị, ứng dụng này đều đưa ra dạng log khác nhau tương ứng với từng nhà cung cấp.

Hệ thống công nghệ thông tin được quản trị bởi nhiều phòng ban như System, Network, Application,...để tổng hợp lại sự kiện tại thời điểm diễn ra sự cố rất khó bởi không có giải pháp chuyên dụng và lưu trữ các sự kiện dài hạn cho việc phân tích sau này, khiến cho một lượng lớn thông tin bị “tràn” dẫn đến việc một số cảnh báo quan trọng có thể bị nhỡ, không được xử lý kịp thời.

Ngoài ra trong thời gian gần đây, các loại hình tấn công kiểu mới như: Advanced Persistent Thread (APT), Zero-day, các loại hình Malware kiểu mới,... ngày càng tinh vi để tránh bị phát hiện và phân tích thì các giải pháp bảo mật truyền thống gần như không có tác dụng.

Vì vậy, giải pháp SIEM ra đời nhằm giải quyết các bài toán phức tạp như trên.[1]

1.1.3. Các thành phần chính của SIEM

Bao gồm 3 thành phần chính:

- Thu thập nhật ký ATTT: Gồm các giao diện có chức năng thu thập nhật ký từ mọi thiết bị. Sau khi tập hợp nó sẽ gửi toàn bộ nhật ký về thành phần phân tích.
- Phân tích và lưu trữ: Tập trung nhật ký và tiến hành phân tích so sánh tương quan. Sau khi thực hiện thuật toán phân tích hệ thống sẽ đưa ra các cảnh báo cần thiết. Thậm chí còn có thể phân tích dữ liệu trong quá khứ.
- Quản trị tập trung: Cung cấp giao diện quản lý tập trung cho toàn bộ hệ thống giám sát an ninh. Hệ thống có sẵn hàng ngàn mẫu báo cáo để có thể sử dụng ngay.

Ngoài ra còn có các thành phần như:

- Thành phần giám sát Network Package ở mức lớp 7 trong mô hình OSI.
- Các module tạo báo cáo (Compliance Report).[1]

1.1.4. Lợi ích của SIEM

Săn tìm và phát hiện các mối đe dọa: Việc sử dụng một SIEM thông minh là chìa khóa, để quản lý các chiến lược, chiến thuật và hoạt động của việc săn lùng mối đe dọa trong bối cảnh mối đe dọa ngày càng tinh vi.

Giảm thời gian phản hồi khi sử dụng tính năng nâng cao trong nhận thức tình huống: SIEM có thể khai thác sức mạnh của trí thông minh về mối đe dọa toàn cầu, để cho phép phát hiện nhanh chóng các sự kiện liên quan đến thông tin liên lạc với các địa chỉ IP đáng ngờ hoặc độc hại. Các đường dẫn tấn công và các tương tác trong quá khứ, có thể được xác định nhanh chóng, giảm thời gian phản hồi để xử lý nhanh hơn các mối đe dọa đối với môi trường.

Tích hợp khả năng hiển thị theo thời gian thực: Tích hợp trên cơ sở hạ tầng bảo mật của bạn mang lại mức độ hiển thị trong thời gian thực về tình hình an ninh của tổ chức bạn

Nguồn lực bảo mật: Các doanh nghiệp có nguồn lực an ninh mạng hạn chế, nhận thấy khả năng quản lý mối đe dọa của SIEM là rất có giá trị.[1]

1.2. Một số giải pháp triển khai SIEM

1.2.1. Giải pháp McAfee SIEM

1.2.1.1. Giới thiệu

Giải pháp SIEM của McAfee sử dụng nhận thức tình huống theo thời gian thực để xác định, hiểu và phản ứng với các mối đe dọa. McAfee SIEM phát hiện và ưu tiên quản lý các sự cố bằng một giải pháp SIEM.

1.2.1.2. Các giải pháp SIEM của McAfee

Đơn giản hóa hoạt động: Các nội dung bảo mật tích hợp và khung tuân thủ được nhúng, giúp đơn giản hóa các hoạt động tuân thủ và phân tích

Biện pháp bảo mật: Cải thiện hiệu quả của tổ chức của bạn, thông qua khả năng hiển thị liên tục, phân tích đưa ra hành động

Phương pháp tiếp cận tích hợp: Một thiết kế có thể mở rộng và tích hợp với hơn ba chục đối tác, hàng trăm nguồn dữ liệu tiêu chuẩn hóa và thông tin tình báo về mối đe dọa trong ngành

1.2.1.3. Các thành phần chính của McAfee SIEM

Enterprise Security Manager (ESM): giải pháp SIEM từng đoạt giải thưởng cung cấp thông tin bảo mật thông minh, nhanh chóng và chính xác

Advanced Correlation Engine (ACE): Liên quan đến dữ liệu được phân tích cú pháp, để xác định mối đe dọa tiềm ẩn, xu hướng và các hoạt động đáng ngờ.

Event Receiver (ERC): Thu thập, phân tích cú pháp và chuẩn hóa dữ liệu bảo mật ban đầu và nhật ký từ 100 nguồn dữ liệu.

Enterprise Log Search (ELS): Săn lùng nhanh hơn bằng cách tìm kiếm hàng tỷ sự kiện trong vài giây và truy cập ngay vào nhật ký để hiểu bối cảnh và tất cả đều được tích hợp trong một bảng điều khiển duy nhất.

Global Threat Intelligence (GTI) for ESM: Được xây dựng cho dữ liệu bảo mật lớn

ESM Cloud: giải pháp SIEM trên đám mây của McAfee cung cấp, có sẵn trong các mẫu được xác định, dễ sử dụng.

1.2.2. Giải pháp IBM Qradar SIEM

1.2.2.1. Tổng quan

IBM QRadar SIEM (Security Information and Event Management – Quản lý sự kiện và bảo mật thông tin) được thiết kế để cung cấp cho các nhóm bảo mật khả năng hiển thị tập trung vào dữ liệu bảo mật toàn doanh nghiệp và hiểu biết sâu sắc về các mối đe dọa ưu tiên cao nhất.

Các tổ chức hiện nay nhận thức rất rõ tầm quan trọng của bảo mật dữ liệu, bao gồm tài sản trí tuệ, bí mật công nghệ, thông tin dữ liệu khách hàng, cũng như đảm bảo hệ thống mạng được vận hành liên tục, tránh gián đoạn trong kinh doanh và các rủi ro hệ thống mạng.

Với thế mạnh về công nghệ của mình, IBM đã cung cấp QRadar SIEM như một giải pháp tổng thể tất cả trong một. Qradar hoạt động như một trung tâm giám sát, bảo mật cho doanh nghiệp, tổ chức với giao diện trực quan, được tích hợp với hàng trăm sản phẩm của IBM cũng như các hãng công nghệ khác.

1.2.2.2. Các dữ liệu được thu thập và phân tích bởi Qrada SIEM

Security events: Từ hệ thống Firewalls, VPN, hệ thống phát hiện và ngăn chặn xâm nhập, cơ sở dữ liệu và hơn thế nữa.

Network events: Từ các thiết bị endpoint như Switches, routers, servers, hosts...

Network activity context: Kiến trúc 7 tầng từ hệ thống mạng và ứng dụng

User or asset context: Dữ liệu từ phạm vi thông tin định danh, thiết bị quản lý truy cập, công cụ quét lỗ hổng bảo mật

Operating system information: Thông tin hệ điều hành, nhà cung cấp, phiên bản cụ thể trong tài nguyên mạng

Application logs: Enterprise resource planning (ERP), quy trình, cơ sở dữ liệu ứng dụng, nền tảng quản lý và vận hành

Threat intelligence: Tức các nguồn như IBM X-Force

1.2.2.3. Lợi ích của IBM Qrada SIEM

Tự động hóa thông tin bảo mật để nhanh chóng phát hiện các mối đe dọa: IPM Qrada SIEM được thiết kế để tự động phân tích và tương quan hoạt động trên nhiều nguồn dữ liệu bao gồm nhật ký, sự kiện, luồng mạng, hoạt động của người dùng, thông tin về lỗ hổng và thông tin về mối đe dọa để xác định các mối đe dọa đã biết và chưa biết

Phát hiện hoạt động mạng, người dùng và ứng dụng bất thường: QRadar mang tới nhiều quy tắc phát hiện hành vi và dị thường như cài đặt mặc định, các nhóm bảo mật cũng có thể tạo quy tắc riêng, cài đặt phát hiện bất thường và tải xuống 160 ứng dụng được xây dựng trước từ IBM Security App Exchange để tăng cường triển khai

Quản lý tốt hơn việc tuân thủ với quy tắc, nội dung và báo cáo được xây dựng trước: QRadar cung cấp tính minh bạch, trách nhiệm và khả năng đo lường quan trọng đối với một tổ chức thành công trong việc đáp ứng các quy định và báo cáo về việc tuân thủ quy định. Giải pháp Khả năng tương quan và tích hợp các nguồn cấp dữ liệu tình báo mối đe dọa mang lại số liệu đầy đủ hơn để báo cáo về rủi ro CNTT cho kiểm

toán viên. Hàng trăm báo cáo được xây dựng trước và các mẫu quy tắc có thể giúp các tổ chức dễ dàng giải quyết các yêu cầu tuân thủ của ngành hơn

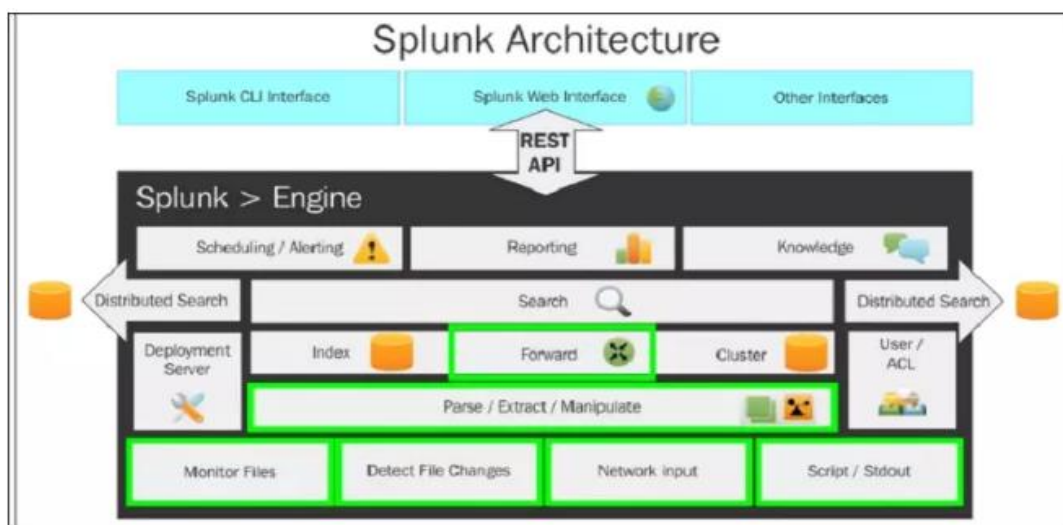
Dễ dàng mở rộng quy mô với nhu cầu thay đổi: Kiến trúc linh hoạt, có thể mở rộng của QRadar được thiết kế để hỗ trợ cả các tổ chức lớn và nhỏ với nhiều nhu cầu khác nhau. Các tổ chức nhỏ hơn có thể bắt đầu với một giải pháp tất cả trong một có thể dễ dàng nâng cấp thành triển khai phân tán khi nhu cầu phát triển

1.2.3. Giải pháp bảo mật Splunk SIEM

1.2.3.1. Tổng quan về giải pháp Splunk SIEM

Splunk là một công cụ SIEM dựa trên phân tích, thu thập, phân tích và so sánh khối lượng lớn dữ liệu mạng và máy móc khác trong thời gian thực. Được quản lý thông qua trình duyệt web, Splunk cung cấp cho các nhóm bảo mật thông tin tình báo có liên quan và có thể hành động được mà họ cần để đối phó hiệu quả với các đe dọa hiệu quả hơn và duy trì một thể trận an ninh chặt chẽ trên quy mô lớn

1.2.3.2. Các thành phần kiến trúc của Splunk



Hình 1.2 Các thành phần kiến trúc của Splunk SIEM

Universal Forwarder: nó là một thành phần nhẹ và đẩy dữ liệu nhật ký vào Splunk forwarder nặng. Nó được cài đặt trên máy chủ ứng dụng hoặc phía máy khách.

Load balancer: nó là bộ cân bằng tải mặc định của Splunk nhưng bạn cũng có thể kết hợp nó với bộ cân bằng tải của mình.

Heavy forward: đây là thành phần nặng cho phép lọc dữ liệu tức là chỉ có thể thu thập các bản ghi lỗi hoặc lâu hơn

Indexer: dùng để lưu trữ cũng như lập chỉ mục dữ liệu nhằm cải thiện hiệu suất tìm kiếm của Splunk.

Search head: nó đang thực hiện vai trò thực hiện báo cáo và giúp thu thập thông tin tình báo.

Deployment service: nó được sử dụng để triển khai cấu hình.

License Manager: nó kiểm tra các chi tiết cấp phép của người dùng. Việc cấp phép được thực hiện dựa trên việc sử dụng và khối lượng.[16]

1.2.3.3. Lợi ích của giải pháp Splunk SIEM

Giám sát an ninh: Splunk liên tục giám sát tất cả các tài nguyên mạng và hoạt động 24/7 để phát hiện hành vi bất thường trước khi nó gây ra mối đe dọa nghiêm trọng cho tổ chức. Sử dụng thông tin mà Splunk cung cấp, các nhóm bảo mật có thể có được cái nhìn chi tiết, theo hướng dữ liệu về hiệu suất, tình trạng và các lỗ hổng của mạng tại bất kỳ thời điểm nào. Hoạt động độc hại hoặc có nguy cơ cao được Splunk phát hiện sẽ tự động cảnh báo cho các bên thích hợp với thông tin ngữ cảnh đầy đủ chi tiết về mối đe dọa

Phát hiện mối đe dọa nâng cao: Giám sát thông minh cơ sở hạ tầng, ứng dụng, người dùng và các tài nguyên mạng khác trên các môi trường cho phép Splunk nắm bắt và ngữ cảnh hóa các mối đe dọa đang hoạt động hoặc hành vi bất thường khi chúng xảy ra trong thời gian thực. Splunk tương quan chéo các bản ghi sự kiện để khám phá các chỉ số về sự thỏa hiệp hoặc các mối quan hệ độc hại để các nhóm bảo mật có thể ngay lập tức xử lý các mối đe dọa tiềm ẩn trước khi có thể gây ra bất kỳ thiệt hại đáng kể nào cho mạng

Phân tích hành vi người dùng: Tận dụng các thuật toán học máy, Splunk chủ động xác định hành vi mạng cũng như tương quan hành vi của người dùng trên các nguồn dữ liệu và môi trường để bắt các mối đe dọa bảo mật khó phát hiện. Những sai lệch so với hoạt động mạng thường xuyên sẽ tự động cảnh báo cho các nhóm bảo mật được chỉ

định để họ có thể nhanh chóng giảm thiểu các mối đe dọa và / hoặc tiến hành điều tra pháp y nhiều bước khi cần thiết

Ứng phó sự cố: Khi phát hiện ra mối đe dọa, các nhóm bảo mật có thể nhanh chóng phản ứng với mức độ tin cậy cao hơn so với công nghệ SIEM cũ. Khung phản hồi thích ứng của Splunk bối cảnh hóa dữ liệu sự kiện trên các môi trường và tự động hóa quy trình phản hồi để các nhà phân tích có thể dễ dàng xác nhận, ưu tiên và xử lý các mối đe dọa bằng thông tin liên quan mà họ cần

Pháp y sự cố: Splunk giám sát và ghi lại các tập dữ liệu thông tin bảo mật khổng lồ được thu thập từ nhiều nguồn mạng khác nhau mỗi ngày. Các nhóm bảo mật có thể sử dụng nguồn dữ liệu dồi dào này để tiến hành các cuộc điều tra pháp y kỹ lưỡng về nguồn gốc của vi phạm hoặc xác thực các mối đe dọa mới xuất hiện để hiểu sâu hơn về hiệu suất của các nỗ lực bảo mật của họ (và thực hiện các cải tiến cho phù hợp).[16]

1.2.3.4. Các sản phẩm Splunk SIEM cung cấp

Splunk Enterprise Security: Nó là một hệ thống SIEM sử dụng dữ liệu do máy tạo ra để có được thông tin chi tiết về hoạt động về các mối đe dọa, lỗ hổng bảo mật, công nghệ bảo mật và thông tin nhận dạng.

Doanh nghiệp Splunk: Nó là một hệ thống thu thập và sau đó phân tích dữ liệu lớn được tạo ra bởi hệ thống cơ sở hạ tầng công nghệ và ứng dụng để có được khả năng hiển thị đầy đủ trên hệ thống bảo mật của doanh nghiệp bạn

Phản hồi thích ứng Splunk: Nó là khuôn khổ cho các hoạt động thích ứng và trong đó, hầu hết các nhà cung cấp bảo mật hàng đầu hợp tác để cải thiện các hoạt động bảo mật và chiến lược phòng thủ mạng.[16]

1.3. Nghiên cứu các bộ dataset phục vụ cho dự đoán bất thường HDFS, HADOOP

1.3.1. Tìm hiểu HADOOP

1.3.1.1. Tổng quan về Hadoop

Hadoop là một framework có mã nguồn mở được viết bằng ngôn ngữ lập trình Java. Hadoop cho phép bạn phát triển được các ứng dụng phân tán và nó có nguồn dữ liệu lớn hoàn toàn miễn phí.

Hadoop đã và đang được phát triển dựa vào các ý tưởng từ Google về các mô hình MapReduce và GFS (Google File System).

Bên cạnh đó thì dựa vào cơ chế streaming nên khi được viết bằng ngôn ngữ Java thì Hadoop vẫn cho phép bạn phát triển các ứng dụng nằm dưới các dạng phân tán dựa vào các loại ngôn ngữ lập trình khác như: C++, Python, Pearl...

1.3.1.2. Cấu trúc Hadoop

1.3.1.2.1. Hadoop Distributed File System (HDFS):

Đây là một trong những hệ thống file phân tán giúp cung cấp cũng như truy cập đến các thông lượng cao dành cho ứng dụng trong việc khai thác dữ liệu. HDFS sử dụng kiến trúc master/slave, trong đó master gồm một NameNode để quản lý hệ thống file metadata và một hay nhiều slave DataNodes để lưu trữ dữ liệu thực tại. Với một tập tin có dạng HDFS đều sẽ được chia thành nhiều khối khác nhau và chúng đều được lưu trữ trong các tập DataNodes. NameNode đều sẽ được định nghĩa là các ánh xạ đến từ những khối từ DataNode. Những loại DataNode này đều điều hành những tác vụ đọc và ghi dữ liệu vào hệ thống file. Ngoài ra, chúng còn có nhiệm vụ quản lý việc tạo, nhân rộng, hủy các khối thông qua những chỉ thị từ NameNode.[15]

1.3.1.2.2. Hadoop MapReduce:

Là một trong những hệ thống dựa trên YARN để có thể thực hiện xử lý song song các tập dữ liệu lớn. MapReduce sẽ bao gồm một single master JobTracker và các slave TaskTracker gay trên mỗi cluster-node. Các Master này đều có nhiệm vụ quản lý được các tài nguyên cũng như theo dõi quá trình tiêu thụ rồi thực hiện lập lịch quản lý các tác vụ trên máy trạm. Hầu hết, các máy slave TaskTracker đều sẽ thực thi được những master theo dạng chỉ định và có thể cung cấp được các thông tin trạng thái tác vụ để cho master có thể theo dõi.

1.3.1.2.3. Hadoop Common:

Đây được xem là một trong những thư viện tiện ích cần thiết để đảm bảo cho các module khác có thể sử dụng. Hầu hết, những thư viện này đều cung cấp cho các hệ thống các file và lớp OS trừu tượng có chứa các mã lệnh Java để có thể khởi động được Hadoop.

1.3.1.2.4. Hadoop YARN:

Là một framework được sử dụng cho việc quản lý cho các tiến trình cũng như tài nguyên của cluster.

1.3.1.3. Hadoop hoạt động như thế nào?

Giai đoạn 1: Một user hay một ứng dụng có thể submit một job lên Hadoop (hadoop job client) với yêu cầu xử lý cùng các thông tin cơ bản:

Nơi lưu (location) dữ liệu input, output trên hệ thống dữ liệu phân tán.

Các java class ở định dạng jar chứa các dòng lệnh thực thi các hàm map và reduce.

Các thiết lập cụ thể liên quan đến job thông qua các thông số truyền vào

Giai đoạn 2: Hadoop job client submit job (file jar, file thực thi) và các thiết lập cho JobTracker. Sau đó, master sẽ phân phối tác vụ đến các máy slave để theo dõi và quản lý tiến trình các máy này, đồng thời cung cấp thông tin về tình trạng và chẩn đoán liên quan đến job-client.

Giai đoạn 3: TaskTrackers trên các node khác nhau thực thi tác vụ MapReduce và trả về kết quả output được lưu trong hệ thống file.[14]

1.3.1.4. Ưu điểm Hadoop:

Hadoop có khả năng thêm nhiều node mới và thay đổi được các cấu hình của chúng một cách dễ dàng.

Các doanh nghiệp không cần phải đầu tư quá nhiều vào phần cứng quá mạnh và đặc biệt khi chạy Hadoop. Nhờ vậy, bạn có thể tiết kiệm được tối đa các chi phí đầu tư ban đầu.

Hadoop có khả năng xử lý được hầu hết những kho dữ liệu có cấu trúc hoặc không có cấu trúc một cách dễ dàng.

Trong suốt quá trình hoạt động thì 1 node trên hệ thống nếu bị lỗi thì nền tảng của Hadoop sẽ có thể tự động di chuyển sang dạng node dự phòng khác. Nhờ vậy mà hệ thống sẽ có thể hoạt động xuyên suốt ổn định hơn.

Hadoop đó là mã nguồn mở, điều này giúp nó tương thích rất nhiều cấu hình và platform khác nhau.

Hadoop có khả năng làm việc cùng với một khối lượng dữ liệu vô cùng lớn.

Hầu hết, các nguồn dữ liệu đều có khả năng được xử lý trong một môi trường dạng phân tán, đồng bộ cũng như được lưu trữ tại nhiều phân cứng khác nhau.

Hadoop sở hữu khả năng băng thông được mọi lưu trữ giữa các phân cứng vật lý của kho dữ liệu có giới hạn cho phép. Chính vì vậy, nó cần được quản lý cũng như nâng cấp sao cho kịp thời nhất. [14]

1.3.2. Tìm hiểu về HDFS

1.3.2.1. Tổng quan về HDFS:

HDFS – Hadoop Distributed File System là framework cho phép tổ chức file trên hệ thống phân tán của Hadoop. HDFS mặc định đi theo Hadoop framework, vậy nếu bạn đã cài đặt thành công Hadoop thì cũng có nghĩa là bạn đã cài đặt được HDFS.

HDFS đóng một vai trò quan trọng trong framework Hadoop cũng như trong hệ sinh thái của Hadoop

1.3.2.2. Kiến trúc của HDFS:

Kiến trúc của HDFS là master / slave. Một HDFS cluster luôn gồm 1 NameNode. NameNode này là 1 master server và nó quản trị mạng lưới hệ thống tập tin cũng như kiểm soát và điều chỉnh truy vấn đến những tập tin khác nhau. Bổ sung cho NameNode có nhiều DataNodes. Luôn có 1 DataNode cho mỗi sever tài liệu. Trong HDFS, 1 tập tin lớn được chia thành 1 hoặc nhiều khối và những khối này được lưu trong 1 tập những DataNodes

1.3.2.3. Ưu điểm của HDFS:

Cho phép lưu trữ phân tán

Cho phép tính toán phân tán và song song

Có khả năng mở rộng theo chiều ngang

1.3.2.4. Đặc trưng của HDFS:

Chi phí: Hadoop có thể cài đặt trên các máy tính có cấu hình phần cứng thông dụng tương tự như cấu hình của một chiếc laptop bình thường. Khi có nhu cầu mở rộng

(bổ sung máy vào hệ), bạn không cần phải chi một số tiền quá lớn, hay nói cách khác thêm một máy vào hệ là rất rẻ

Có thể lưu trữ bigdata vô tư: Đương nhiên, vai trò của HDFS vốn là để lưu trữ bigdata, vì vậy bạn cứ yên tâm HDFS có thể lưu trữ được một lượng dữ liệu khổng lồ (Terabytes & petabytes dữ liệu) và đa dạng (có cấu trúc, bán cấu trúc, không có cấu trúc).

Dữ liệu có độ tin cậy cao và có khả năng khắc phục sau lỗi tốt: Dữ liệu lưu trữ trong HDFS sẽ được nhân bản thành nhiều phiên bản (mặc định là 3 và có thể cấu hình được). Nghĩa là khi bạn lưu một file nặng 1GB thì HDFS sẽ dành ra 3GB để lưu trữ file đó. Vì vậy mà khi có một máy bị lỗi, thì dữ liệu vẫn còn bản backup tại máy khác và đảm bảo được tính toàn vẹn (độ tin cậy cao). Khi máy bị lỗi được khắc phục, dữ liệu sẽ dễ dàng được khôi phục lại từ các máy khác.

Tính chính xác cao: Thể hiện việc dữ liệu bạn lấy ra có chính xác như những gì bạn đã lưu không. Dữ liệu lưu trữ trong HDFS sẽ thường xuyên được kiểm tra, nếu có phát hiện sai lệch thì sẽ tự động được khôi phục bằng các bản sao.

Có Throughput cao: Throughput là số lượng công việc hoàn thiện trong một đơn vị thời gian. Nó thể hiện tốc độ xử lý dữ liệu, truy cập dữ liệu. Bạn còn nhớ cái ví dụ về 10 máy tính xử lý song song chỉ mất có 6 phút mà mình đề cập ở trên không, đó chính là một ví dụ thể hiện cho tính Throughput cao đó.

Xử lý dữ liệu “tại chỗ”: Cách xử lý dữ liệu truyền thống là dữ liệu cần xử lý sẽ được gửi tới một ứng dụng nào đó trong máy tính để xử lý. Tuy nhiên với đặc thù của big data là khối lượng dữ liệu rất lớn nên việc di chuyển dữ liệu tới ứng dụng để xử lý sẽ tốn nhiều thời gian và tài nguyên máy. Vì vậy thay vì mang dữ liệu tới “chỗ xử lý”, thì HDFS sẽ mang “chỗ xử lý” tới gặp dữ liệu.[15]

1.4. Tìm hiểu một số hình thức tấn công trong Top 10 OWASP

1.4.1. SQL Injection

1.4.1.1. SQL Injection là gì?

Là một kỹ thuật lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng. Kẻ tấn công có thể chèn các đoạn mã để lấy cắp dữ liệu và chiếm quyền kiểm soát trình duyệt của người dùng.

1.4.1.2. Cách thức hoạt động

Được thực hiện bằng cách chèn thêm một đoạn SQL để làm sai lệnh đi câu truy vấn ban đầu, từ đó có thể khai thác dữ liệu từ database.

1.4.1.3. Tác hại

Ăn cắp dữ liệu trang web hoặc hệ thống

Thay đổi dữ liệu hệ thống.

Xóa dữ liệu nhạy cảm và quan trọng trên hệ thống.

Sửa đổi cấu trúc của cơ sở dữ liệu.

Kiểm soát máy chủ cơ sở dữ liệu và thực thi lệnh theo ý muốn.

1.4.1.4. Cách phòng chống SQL Injection

Luôn kiểm tra, ràng buộc dữ liệu nhập vào: Dữ liệu phải được xác thực trước khi sử dụng trong các câu lệnh SQL.

Dùng Regular Expression để loại bỏ đi các ký tự lạ hoặc các ký tự không phải là số.

Sao lưu dữ liệu thường xuyên: Dữ liệu cần được sao lưu thường xuyên để trong trường hợp xấu nhất là bị tin tặc xóa thì doanh nghiệp vẫn có thể khôi phục lại.

Phân quyền truy cập rõ ràng: Việc cho phép mọi tài khoản đều được truy cập vào cơ sở dữ liệu tiềm ẩn nhiều rủi ro. Vì vậy hãy chỉ định một số tài khoản nhất định có quyền kết nối với cơ sở dữ liệu.[2]

1.4.2. XSS (Cross Site Scripting)

1.4.2.1 XSS (Cross Site Scripting) là gì ?

Là một lỗ hổng khá phổ biến, kẻ tấn công chèn các đoạn mã JavaScript vào ứng dụng web. Khi đầu vào này không được lọc, chúng sẽ thực thi mã độc trên trình duyệt của người dùng. Kẻ tấn công có thể lấy được cookie của người dùng trên hệ thống hoặc lừa người dùng đến các trang web độc hại.

1.4.2.2 Cách thức hoạt động

Kẻ tấn công truyền một payload vào cơ sở dữ liệu của trang web bằng cách gửi một biểu mẫu kèm theo một số JavaScript độc hại. Nạn nhân gửi yêu cầu đến trang web.

Trang web hiển thị trên trình duyệt của nạn nhân với một phần nội dung HTML chứa payload của kẻ tấn công.

Trình duyệt của nạn nhân sẽ thực thi tập lệnh độc hại bên trong HTML

1.4.2.3 Tác hại

Mạo danh người dùng.

Độc dữ liệu người dùng.

Lấy cắp thông tin đăng nhập.

Tiêm trojan vào trang web.

1.4.2.4 Cách phòng chống XSS

Data Validation (Xác thực đầu vào): Mọi thứ do người dùng nhập phải được xác thực chính xác vì thông tin Input của người dùng có thể tìm đường đến Output. Việc xác thực dữ liệu có thể được đặt tên là cơ sở để đảm bảo tính bảo mật của hệ thống.

Filtering (Lọc đầu vào): Ý tưởng lọc là tìm kiếm các từ khóa nguy hiểm trong mục nhập của người dùng và xóa chúng hoặc thay thế chúng bằng các chuỗi trống.

Escaping (Kí tự Escape): Một phương pháp phòng ngừa khác có thể là ký tự Escape. Trong thực tế này, các ký tự thích hợp đang được thay đổi bằng các mã đặc biệt.[2]

CHƯƠNG 2: TÌM HIỂU CÁC THUẬT TOÁN MÁY HỌC ỨNG DỤNG TRONG PHÁT HIỆN BẤT THƯỜNG

2.1. Support Vector Machine (SVM)

2.1.1. Tổng quan SVM

SVM là một thuật toán học máy nổi tiếng được sử dụng để giải quyết bài toán phân lớp dữ liệu.

Thuật toán SVM ban đầu được phát minh bởi Vladimir N. Vapni còn thuật toán SVM tiêu chuẩn hiện nay được đề xuất bởi Vladimir N. Vapnik và Corinna Cortes vào năm 1995.

SVM đã được áp dụng rất thành công trong việc giải quyết các vấn đề của thực tế xã hội như nhận dạng văn bản, nhận dạng hình ảnh, nhận dạng chữ viết tay, phân loại thư rác điện tử, phát hiện xâm nhập mạng,...

Ban đầu bài toán SVM được viết cho bài toán phân lớp nhị phân.[10]

2.1.2. Ý tưởng của SVM

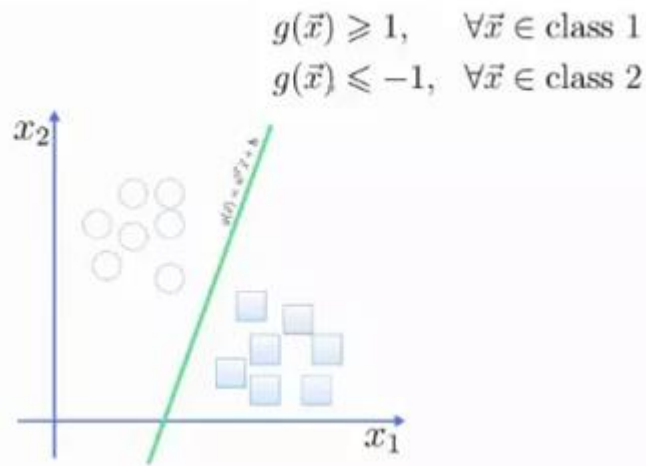
Cho $X=\{x_i\}$ là tập hợp các vector trong không gian R^D (R mũ D) và x_i thuộc 1 trong 2 lớp $y_i=1$ hoặc $y_i=-1$. Ta có tập điểm dữ liệu huấn luyện được biểu diễn như sau : $\{x_i, y_i\}$ với $i=1, \dots, n$ với $y_i \in \{-1, 1\}$, n là số điểm dữ liệu huấn luyện.

Giả sử rằng dữ liệu là phân tách tuyến tính, nghĩa là có thể tìm được ít nhất một đường thẳng trên đồ thị của x_1 và x_2 phân tách 2 lớp khi $D=2$ và một siêu phẳng trên đồ thị của $x_1, x_2, x_3, \dots, x_D$ phân tách 2 lớp khi $D>2$.

Mục tiêu của SVM là xây dựng một siêu phẳng giữa 2 lớp sao cho khoảng cách từ nó tới các điểm gần siêu phẳng nhất của 2 lớp là cực đại. Siêu phẳng có thể được miêu tả bởi phương trình : $\omega \cdot x + b = 0$ trong đó :

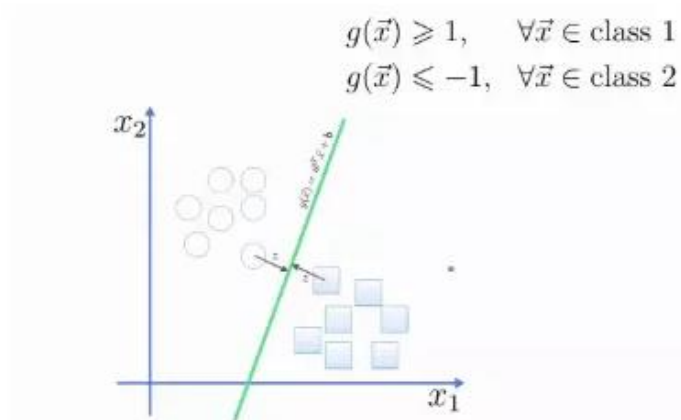
- " \cdot " là phép nhân vô hướng véctơ.
- " ω " là véctơ pháp tuyến của siêu phẳng.
- $b / \|\omega\|$ là khoảng cách vuông góc từ siêu phẳng đến gốc tọa độ

Khi đó vector hỗ trợ (support vector) là những điểm dữ liệu gần siêu phẳng phân tách nhất.



Hình 2.1 Đồ thị miêu tả bài toán phân 2 lớp SVM

Gọi z là khoảng cách từ các điểm gần mặt phân cách nhất của 2 lớp đến mặt phân cách (hay còn gọi là lề)



Hình 2.2 Mô tả lề trong đồ thị của bài toán phân 2 lớp SVM

Theo hình học không gian thì $z = |g(x)| / \|\omega\|$

- Với lớp 1 thì $g(x) \geq 1 \Rightarrow z = 1 / \|\omega\|$. Tương tự như vậy với lớp 2
- Để z là cực đại thì $\|\omega\|$ phải là cực tiểu \Rightarrow bài toán trở thành : Tìm min $\|\omega\|$ thỏa mãn $(x_i \cdot \omega + b) - 1 \geq 0 \forall x_i$.
- Khi đã tìm được ω_0, b_0 thỏa mãn điều kiện trên, một mẫu x' sẽ được phân lớp.

- Người ta chỉ ra rằng nếu các vector huấn luyện được phân tách mà không có lỗi bởi một siêu phẳng thì xác suất lỗi mắc phải trên một mẫu kiểm tra được giới hạn bởi tỉ lệ giữa giá trị kì vọng của số lượng vector hỗ trợ và số lượng vector huấn luyện:
- $E[\text{Pr}(\text{error})] = E[\text{số vector hỗ trợ}] / \text{số vector huấn luyện}$ [10]

2.1.3. Ưu điểm của SVM

Là một kĩ thuật phân lớp khá phổ biến, SVM thể hiện được nhiều ưu điểm trong số đó có việc tính toán hiệu quả trên các tập dữ liệu lớn. Có thể kể thêm một số Ưu điểm của phương pháp này như:

Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.

Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.

Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.[10]

2.1.4. Nhược điểm của SVM

Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi.

Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào. Tuy nhiên hiệu quả của việc phân lớp có thể được xác định dựa vào khái niệm margin từ điểm dữ liệu mới đến siêu phẳng phân lớp mà chúng ta đã bàn luận ở trên.

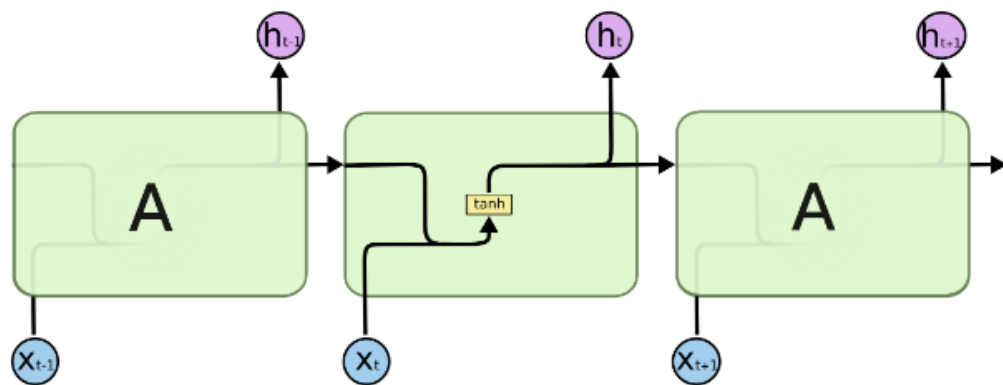
2.2. Long Short Term Memory (LSTM)

2.2.1. Tổng quan LSTM

Mạng trí nhớ ngắn hạn định hướng dài hạn còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN có khả năng học được sự phụ thuộc trong dài hạn (long-term dependencies) được giới thiệu bởi Hochreiter & Schmidhuber (1997). Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay.

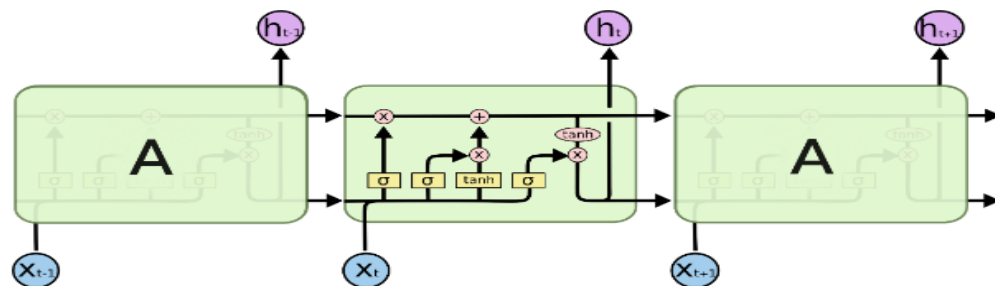
LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn giữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.

Một mạng RNN tiêu chuẩn sẽ có kiến trúc rất đơn giản chẳng hạn như đối với kiến trúc gồm một tầng ẩn là hàm tanh như bên dưới.



Hình 2.3 Sự lặp lại kiến trúc module trong mạng RNN chứa 1 tầng ẩn

LSTM cũng có một chuỗi dạng như thế nhưng phần kiến trúc lặp lại có cấu trúc khác biệt hơn. Thay vì chỉ có một tầng đơn, chúng có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt.



Hình 2.4 Sự lặp lại kiến trúc module trong mạng LTSM chứa 4 tầng ẩn

Các kí hiệu:



Hình 2.5 Ký hiệu biểu diễn kiến trúc

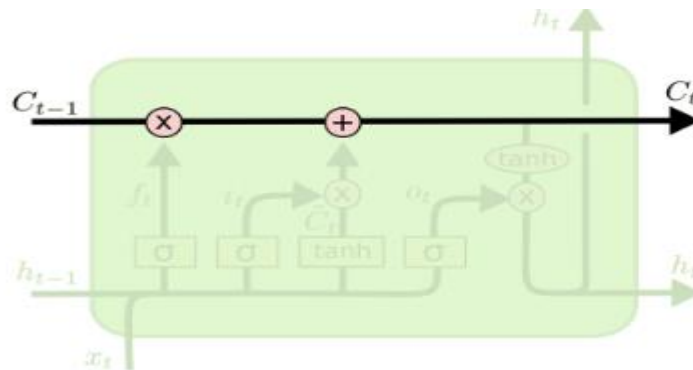
Neural Network layer: Hàm activation mà mạng nơ ron sử dụng để học trong tầng ẩn, thông thường là các hàm phi tuyến sigmoid và tanh

Pointwise Operation: Biểu diễn một toán tử đối với véc tơ như phép cộng véc tơ, phép nhân vô hướng các véc tơ

Concatenate: Phép chập kết quả trong khi kí hiệu 2 đường thẳng rẽ nhánh thể hiện cho nội dung véc tơ trước đó được sao chép để đi tới một phần khác của mạng nơ ron.[11]

2.2.2. Ý tưởng LSTM

Ý tưởng chính của LSTM là thành phần ô trạng thái (cell state) được thể hiện qua đường chạy ngang qua đỉnh đồ thị như hình vẽ bên dưới:

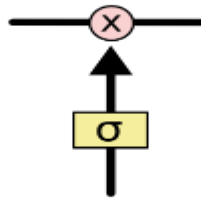


Hình 2.6 Đường đi của ô trạng thái (Cell state) trong mạng LSTM

Ô trạng thái là một dạng băng chuyền chạy thẳng xuyên suốt toàn bộ chuỗi với chỉ một vài tương tác tuyến tính nhỏ giúp cho thông tin có thể truyền dọc theo đồ thị mạng nơ ron ổn định.

LSTM có khả năng xóa và thêm thông tin vào ô trạng thái và điều chỉnh các luồng thông tin này thông qua các cấu trúc gọi là cổng.

Cổng là cơ chế đặc biệt để điều chỉnh luồng thông tin đi qua. Chúng được tổng hợp bởi một tầng ẩn của hàm activation sigmoid và với một toán tử nhân như đồ thị.



Hình 2.7 Một cổng của hàm sigmoid trong LSTM

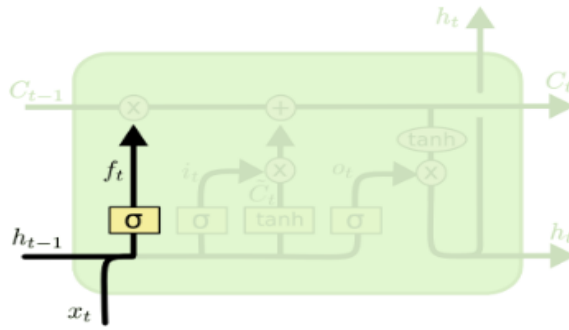
Hàm sigmoid sẽ cho đầu ra là một giá trị xác suất nằm trong khoảng từ 0 đến 1, thể hiện rằng có bao nhiêu phần thông tin sẽ đi qua cổng. Giá trị bằng 0 ngụ ý rằng không cho phép thông tin nào đi qua, giá trị bằng 1 sẽ cho toàn bộ thông tin đi qua.

Một mạng LSTM sẽ có 3 cổng có kiến trúc dạng này để bảo vệ và kiểm soát các ô trạng thái.[11]

2.2.3. Thứ tự các bước của LSTM

Bước đầu tiên trong LSTM sẽ quyết định xem thông tin nào chúng ta sẽ cho phép đi qua ô trạng thái (cell state). Nó được kiểm soát bởi hàm sigmoid trong một tầng gọi là tầng quên (forget gate layer).

Đầu tiên nó nhận đầu vào là 2 giá trị h_{t-1} và x_t và trả về một giá trị nằm trong khoảng 0 và 1 cho mỗi giá trị của ô trạng thái C_{t-1} . Nếu giá trị bằng 1 thể hiện ‘giữ toàn bộ thông tin’ và bằng 0 thể hiện ‘bỏ qua toàn bộ chúng’.

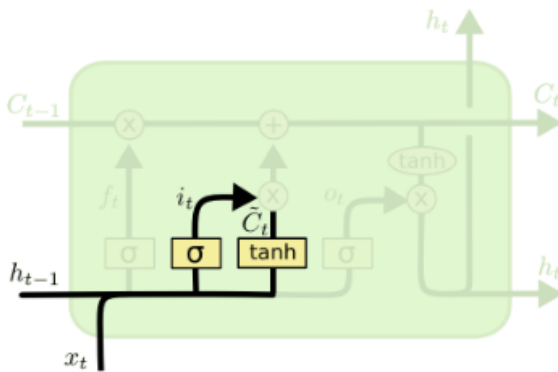


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Hình 2.8 Tầng cổng quên (Forget gate layer)

Bước tiếp theo chúng ta sẽ quyết định loại thông tin nào sẽ được lưu trữ trong ô trạng thái. Bước này bao gồm 2 phần. Phần đầu tiên là một tầng ẩn của hàm sigmoid được gọi là tầng cổng vào (input gate layer) quyết định giá trị bao nhiêu sẽ được cập nhật. Tiếp theo, tầng ẩn hàm tanh sẽ tạo ra một véc tơ của một giá trị trạng thái mới \tilde{C}_t mà có thể được thêm vào trạng thái. Tiếp theo kết hợp kết quả của 2 tầng này để tạo thành một cập nhật cho trạng thái.

Trong ví dụ của mô hình ngôn ngữ, chúng ta muốn thêm loại của một chủ ngữ mới vào ô trạng thái để thay thế phần trạng thái cũ muốn quên đi.



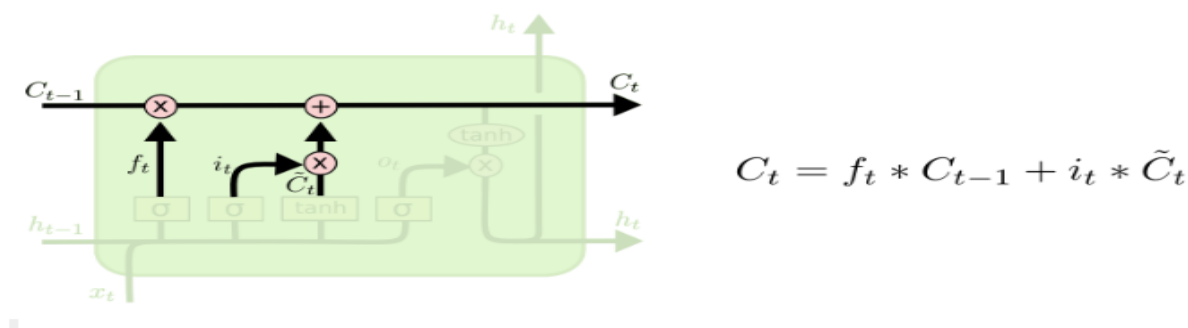
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Hình 2.9 Cập nhật giá trị cho ô trạng thái bằng cách kết hợp 2 kết quả từ tầng cổng vào và tầng ẩn hàm tanh

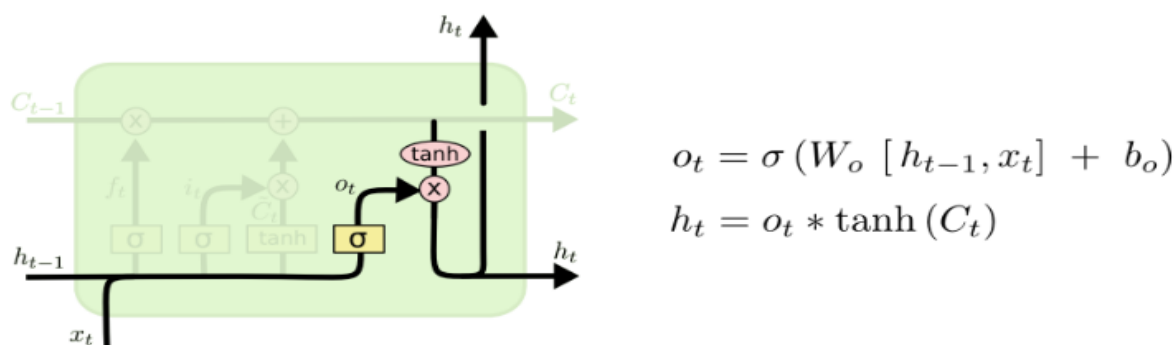
Đây là thời điểm để cập nhật một ô trạng thái cũ, C_{t-1} sang một trạng thái mới C_t . Những bước trước đó đã quyết định làm cái gì, và tại bước này chỉ cần thực hiện nó.

Chúng ta nhân trạng thái cũ với f_t tương ứng với việc quên những thứ quyết định được phép quên sớm. Phần tử đề cử $i_t * \tilde{C}_t$ là một giá trị mới được tính toán tương ứng với bao nhiêu được cập nhật vào mỗi giá trị trạng thái.



Hình 2.10 Ô trạng thái mới

Bước cuối cùng cần quyết định xem đầu ra sẽ trả về bao nhiêu. Kết quả ở đầu ra sẽ dựa trên ô trạng thái, nhưng sẽ là một phiên bản được lọc. Đầu tiên, chúng ta chạy qua một tầng sigmoid nơi quyết định phần nào của ô trạng thái sẽ ở đầu ra. Sau đó, ô trạng thái được đưa qua hàm tanh (để chuyển giá trị về khoảng -1 và 1) và nhân nó với đầu ra của một cổng sigmoid, do đó chỉ trả ra phần mà chúng ta quyết định.[11]



Hình 2.11 Điều chỉnh thông tin ở đầu ra thông qua hàm tanh

2.2.4. Kết luận

LSTM là một bước đột phá lớn mà ở đó chúng ta đã khắc phục được những hạn chế ở RNN đó là khả năng phụ thuộc dài hạn. Một số kỹ thuật học Attention gần đây được kết hợp với LSTM đã tạo ra những kết quả khá bất ngờ trong các tác vụ dịch máy cũng như phân loại nội dung, trích lọc thông tin,... Các mô hình dịch máy của google

đã ứng dụng kiểu kết hợp này trong các bài toán dịch thuật của mình và đã cải thiện được nội dung bản dịch một cách đáng kể.

2.3. Recurrent Neural Network (RNN)

2.3.1. Tổng quan RNN

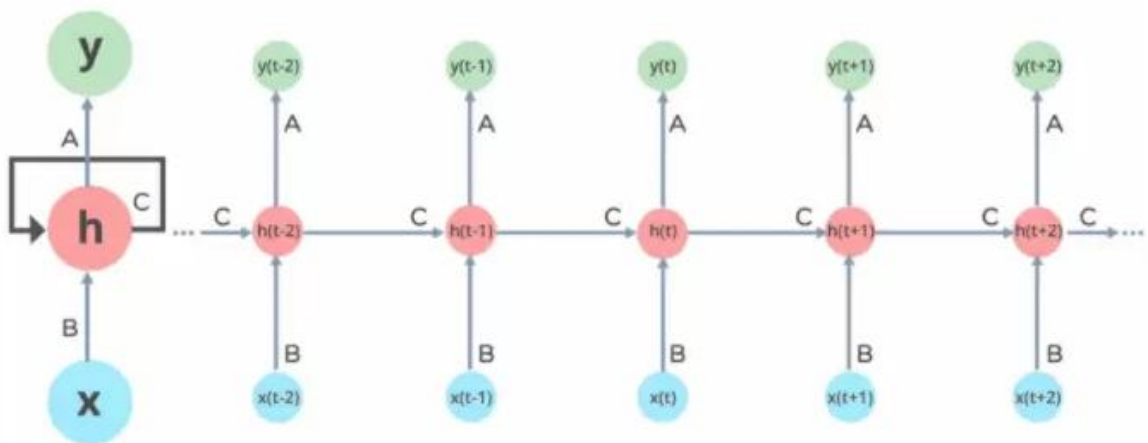
Recurrent Neural Network (RNN) là một loại mạng nơ-ron nhân tạo chủ yếu được sử dụng trong nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên (NLP). RNN được sử dụng trong học tập sâu và trong việc phát triển các mô hình bắt chước hoạt động của các tế bào thần kinh trong não người.

Mạng lặp lại được thiết kế để nhận dạng các mẫu trong chuỗi dữ liệu, chẳng hạn như văn bản, bộ gen, chữ viết tay, lời nói và dữ liệu chuỗi thời gian số phát ra từ cảm biến, thị trường chứng khoán và các cơ quan chính phủ

Recurrent Neural Network trông tương tự như mạng nơ-ron truyền thống ngoại trừ một trạng thái bộ nhớ được thêm vào các nơ-ron. Việc tính toán là bao gồm một bộ nhớ đơn giản

Recurrent Neural Network là một loại thuật toán hướng đến học sâu, theo cách tiếp cận tuần tự. Trong mạng nơ-ron, chúng ta luôn giả định rằng mỗi đầu vào và đầu ra phụ thuộc vào tất cả các lớp khác. Các loại mạng nơ-ron này được gọi là mạng tái phát vì chúng thực hiện tuần tự các phép tính toán học.[12]

2.3.2. Phương thức hoạt động RNN



Hình 2.12 Hoạt động của RNN

Trong mạng Neural RNN, thông tin chuyển qua một vòng lặp đến lớp ẩn giữa.

Lớp đầu vào 'x' nhận đầu vào cho mạng nơ-ron và xử lý nó và chuyển nó vào lớp giữa.

Lớp giữa 'h' có thể bao gồm nhiều lớp ẩn, mỗi lớp có chức năng kích hoạt và trọng số và thành kiến riêng. Nếu bạn có một mạng nơron trong đó các tham số khác nhau của các lớp ẩn khác nhau không bị ảnh hưởng bởi lớp trước đó, tức là: mạng nơron không có bộ nhớ, thì bạn có thể sử dụng mạng nơron tuần hoàn.

RNN sẽ chuẩn hóa các chức năng kích hoạt và trọng số và độ lệch khác nhau để mỗi lớp ẩn có các tham số giống nhau. Sau đó, thay vì tạo nhiều lớp ẩn, nó sẽ tạo một lớp và lặp lại nhiều lần theo yêu cầu.[12]

2.3.3. Ứng dụng của RNN

RNN có nhiều công dụng khi dự đoán tương lai. Trong ngành tài chính, RNN có thể giúp dự đoán giá cổ phiếu hoặc dấu hiệu của xu hướng thị trường chứng khoán (tức là tích cực hoặc tiêu cực).

RNN được sử dụng cho ô tô tự lái vì nó có thể tránh tai nạn ô tô bằng cách đoán trước tuyến đường của xe.

RNN được sử dụng rộng rãi trong chú thích hình ảnh, phân tích văn bản, dịch máy và phân tích tình cảm. Ví dụ, người ta nên sử dụng đánh giá phim để hiểu cảm giác mà khán giả cảm nhận sau khi xem phim. Tự động hóa nhiệm vụ này rất hữu ích khi công ty điện ảnh không thể có nhiều thời gian hơn để xem xét, củng cố, dán nhãn và phân tích các bài đánh giá. Máy có thể thực hiện công việc với mức độ chính xác cao hơn.[12]

2.3.4. Phân loại RNN

One-to-one: Đây còn được gọi là mạng Neural thuần túy. Nó xử lý một kích thước cố định của đầu vào với kích thước cố định của đầu ra, nơi chúng độc lập với thông tin đầu ra trước đó.

Ví dụ: Phân loại ảnh.

One-to Many: Nó xử lý một kích thước cố định của thông tin làm đầu vào cung cấp một chuỗi dữ liệu làm đầu ra.

Ví dụ: Image Captioning lấy hình ảnh làm đầu vào và đầu ra một câu từ.

Many-to-One: Nó lấy một chuỗi thông tin làm đầu vào và đầu ra với kích thước cố định của đầu ra.

Ví dụ: phân tích tình cảm trong đó bất kỳ câu nào được phân loại là thể hiện tình cảm tích cực hoặc tiêu cực.

Many-to-Many: Nó lấy Chuỗi thông tin làm đầu vào và xử lý các đầu ra lặp lại dưới dạng Chuỗi dữ liệu.

Ví dụ: Dịch máy, trong đó RNN đọc bất kỳ câu nào bằng tiếng Anh và sau đó xuất ra câu đó bằng tiếng Pháp.

Bidirectional Many-to-Many: Đầu vào và đầu ra trình tự được đồng bộ hóa. Lưu ý rằng trong mọi trường hợp không có ràng buộc nào được chỉ định trước đối với trình tự độ dài bởi vì phép biến đổi lặp lại (màu xanh lá cây) là cố định và có thể được áp dụng nhiều lần tùy thích[12]

2.3.5. Hạn chế của RNN

RNN có nhiệm vụ đưa thông tin kịp thời. Tuy nhiên, việc truyền tất cả các thông tin này là một việc khá khó khăn khi bước thời gian quá dài. Khi một mạng có quá nhiều lớp sâu, nó sẽ trở nên không thể kiểm tra được. Vấn đề này được gọi là: vấn đề gradient biến mất.

Nếu chúng ta nhớ, mạng nơ-ron cập nhật việc sử dụng trọng số của thuật toán giảm độ dốc. Gradient nhỏ dần khi mạng tiến xuống các lớp thấp hơn.

Gradient không đổi, có nghĩa là không có không gian để cải thiện. Mô hình học hỏi từ sự thay đổi trong gradient của nó; sự thay đổi này ảnh hưởng đến đầu ra của mạng [12]

2.4. Convolutional Neural Network (CNNs)

2.4.1. Tổng quan CNNs

Mạng nơ-ron tích hợp (CNN hoặc ConvNet), là một kiến trúc mạng dành cho học sâu, học trực tiếp từ dữ liệu, loại bỏ nhu cầu trích xuất tính năng thủ công.

CNN đặc biệt hữu ích để tìm các mẫu trong hình ảnh để nhận dạng các đối tượng, khuôn mặt và cảnh. Chúng cũng có thể khá hiệu quả để phân loại dữ liệu không phải hình ảnh như dữ liệu âm thanh, chuỗi thời gian và tín hiệu.

Các ứng dụng yêu cầu nhận dạng vật thể và tầm nhìn máy tính - chẳng hạn như xe tự lái và ứng dụng nhận dạng khuôn mặt - phụ thuộc rất nhiều vào CNN.[13]

2.4.2. Điều gì khiến CNN trở nên hữu ích như vậy?

Sử dụng CNN để học sâu là phổ biến do ba yếu tố quan trọng:

- CNN loại bỏ nhu cầu trích xuất tính năng thủ công - các tính năng được CNN học trực tiếp.
- CNN tạo ra kết quả nhận dạng chính xác cao.
- CNN có thể được đào tạo lại cho các nhiệm vụ nhận dạng mới, cho phép bạn xây dựng trên các mạng đã có từ trước.

2.4.3. Ứng dụng của CNN

CNN cung cấp một kiến trúc tối ưu để khám phá và tìm hiểu các tính năng chính trong dữ liệu hình ảnh và chuỗi thời gian. CNN là một công nghệ quan trọng trong các ứng dụng như:

Hình ảnh y tế : CNN có thể kiểm tra hàng nghìn báo cáo bệnh lý để phát hiện trực quan sự hiện diện hoặc vắng mặt của tế bào ung thư trong hình ảnh.

Xử lý âm thanh : Tính năng phát hiện từ khóa có thể được sử dụng trong bất kỳ thiết bị nào có micro để phát hiện khi một từ hoặc cụm từ nhất định được nói - ('Hey Siri!'). CNN có thể học và phát hiện chính xác từ khóa trong khi bỏ qua tất cả các cụm từ khác bất kể môi trường.

Phát hiện biển báo dừng : Lái xe tự động dựa vào CNN để phát hiện chính xác sự hiện diện của biển báo hoặc vật thể khác và đưa ra quyết định dựa trên kết quả đầu ra.

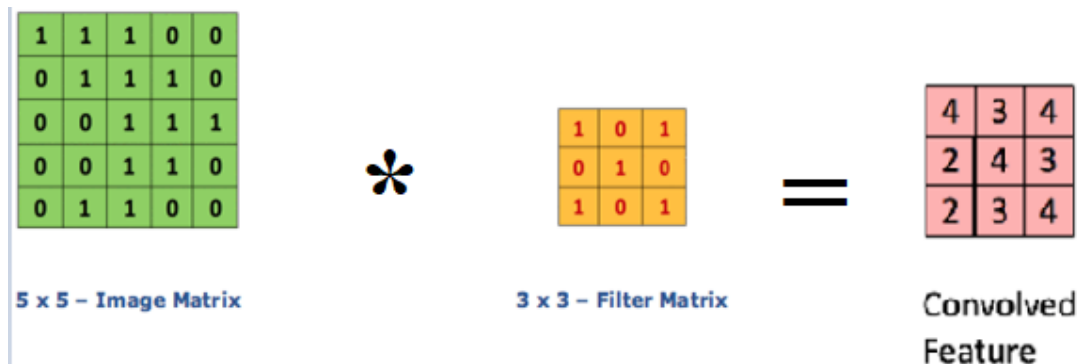
Tạo dữ liệu tổng hợp : Sử dụng mạng đối phương chung (GAN) , hình ảnh mới có thể được tạo ra để sử dụng trong các ứng dụng học sâu bao gồm nhận dạng khuôn mặt và lái xe tự động.[13]

2.4.4. Tìm hiểu tính năng, lớp và phân loại

Một mạng nơ-ron phức tạp có rất nhiều lớp phân loại, mỗi lớp sẽ học cách phát hiện các đặc điểm khác nhau của một hình ảnh. Các bộ lọc (Kernels) được áp dụng cho mỗi hình ảnh đào tạo ở các độ phân giải khác nhau và đầu ra của mỗi hình ảnh được kết hợp được sử dụng làm đầu vào cho lớp tiếp theo. Các bộ lọc có thể bắt đầu với các đặc điểm rất đơn giản, chẳng hạn như độ sáng, cho đến các đặc điểm phức tạp mà chỉ từng đối tượng có.

Gọi là mạng nơ-ron nên CNN cũng sẽ có các thành phần gồm một lớp dữ liệu đầu vào, một lớp dữ liệu đầu ra và nhiều lớp dữ liệu ẩn ở giữa. Các lớp này thực hiện các hoạt động thay đổi dữ liệu với mục đích tìm hiểu các tính năng cụ thể cho dữ liệu. Ba trong số các lớp phổ biến nhất là: tích chập (convolution), hàm phi tuyến (ReLU) và gộp lại(pooling).

Tích chập (Convolution) là lớp trích xuất các tính năng từ hình ảnh dữ liệu đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vuông nhỏ của dữ liệu đầu vào. Đây luôn là những thông tin chính xác nhất của dữ liệu. Ví dụ : nếu dữ liệu đầu vào là một ma trận với kích thước $H \times W$, lớp tích chập sẽ dùng một bộ lọc (filter) với kích thước $h \times w$ để nhân từng phần tử với ma trận của dữ liệu để cho ra một mẫu đặc trưng (feature map) bằng cách trượt đều trên ma trận, độ lớn của feature map lúc này sẽ là $(H-h+1) \times (W-w+1)$.



Hình 2.13 Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau

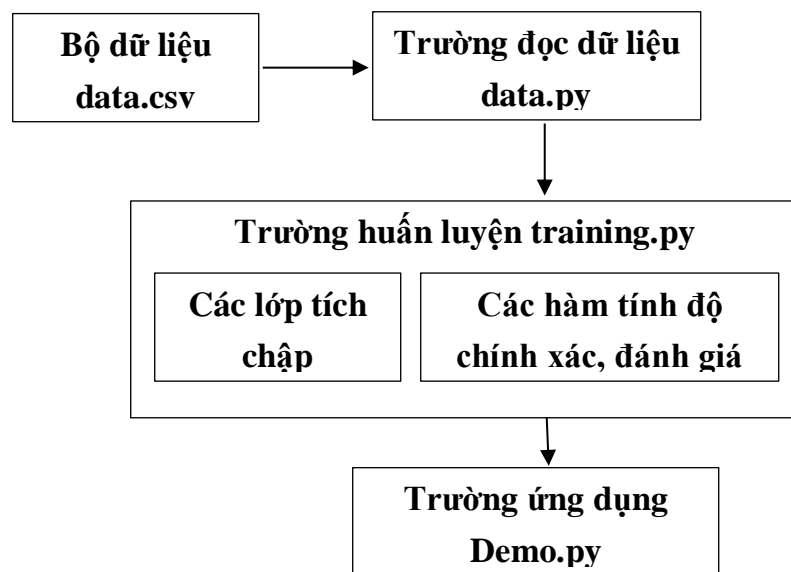
Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể thực hiện các hoạt động khác nhau như phát hiện cạnh, làm mờ và làm sắc nét bằng cách áp dụng các bộ lọc.

Hàm phi tuyến (ReLU) cho phép đào tạo nhanh hơn và hiệu quả hơn bằng cách ánh xạ các giá trị không âm. Có một số hàm phi tuyến khác như tanh, sigmoid cũng có thể được sử dụng thay cho ReLU. Hầu hết người ta thường dùng ReLU vì nó có hiệu suất tốt.

Gộp lại (pooling) giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu xuống làm giảm kích thước của mỗi ma trận nhưng vẫn giữ lại thông tin quan trọng. Có các loại như max pooling lấy phần tử lớn nhất từ ma trận đối tượng, average pooling hoặc lấy trung bình, sum pooling thì lấy tổng tất cả các phần tử.[13]

2.4.5. Thực hiện thử nghiệm thuật toán

Ta sẽ khởi tạo một mô hình phát hiện tấn công mã độc XSS với dữ liệu đầu vào là một payload XSS đã được tổng hợp, sử dụng ngôn ngữ Python để thực hiện quá trình training dữ liệu trên cho máy, thư viện hỗ trợ tensorflow giúp tính toán với các hàm có sẵn.



Hình 2.14 Sơ đồ thử nghiệm thuật toán CNNs

Tóm tắt các bước thực hiện thử nghiệm thuật toán CNNs:

- Bước 1: Tiền xử lý:

Đưa File Data.csv vào, tiến hành thao tác đọc File dữ liệu và chuyển dữ liệu sang dạng mã ASCII để máy có thể đọc được

- Bước 2: Xử lý và chạy máy học:

Tạo hàm đánh giá hiệu suất với tỉ lệ tập huấn luyện và tập thử nghiệm là 0.2, số lần chạy máy học là 150

Tạo mô hình mạng nơ-ron với 11 lớp

Thực hiện cho máy học, tính toán và dự đoán độ an toàn của dữ liệu

- Bước 3: Đưa ra kết quả sau khi chạy máy học

- Nếu như kết quả có độ tin cậy lớn hơn 50% thì trả về là giá trị không an toàn (XSS), ngược lại trả về là giá trị an toàn.

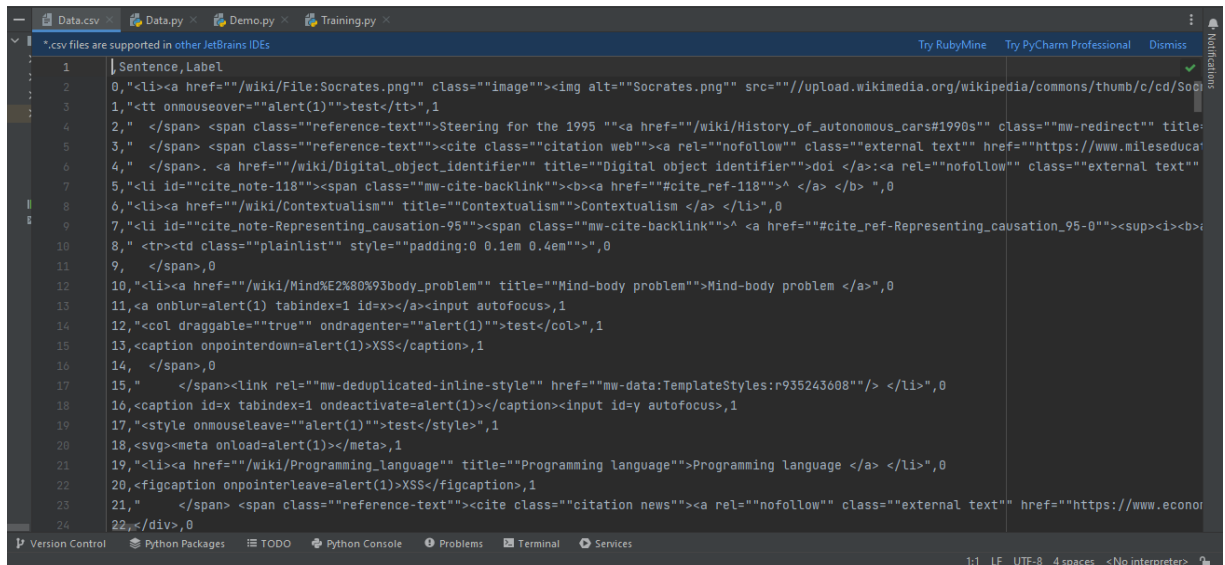
File data.csv đưa vào chạy máy học gồm có 3 field chính :

- Field 1(Stt) : trường chứa số thứ tự
- Field 2(Sentence): trường chứa các giá trị cần học máy
- Field 3(Label) : trường chứa nhãn, cho biết giá trị nào có dấu hiệu tấn công và giá trị nào là an toàn

	A	B	C	D	E	F	G	H	I	J
1		Sentence	Label							
2	0	<img alt	0							
3	1	<tt onmouseover="alert(1)">test</tt>	1							
4	2	 Steering for the 1995	0							
5	3	 <cite class="citation w	0							
6	4	. <a href="/wiki/Digital_object_identifier" title="Dig	0							
7	5	<li id="cite_note-118"><	0							
8	6	Cc	0							
9	7	<li id="cite_note-Representing_causation-95"><span class="i	0							
10	8	<tr><td class="plainlist" style="padding:0 0.1em 0.4em">	0							
11	9		0							
12	10	<a href="/wiki/Mind%E2%80%93body_problem" title="M	0							
13	11	<input autofocus>	1							
14	12	<col draggable="true" ondragenter="alert(1)">test</col>	1							
15	13	<caption onpointerdown=alert(1)>XSS</caption>	1							
16	14		0							
17	15	<link rel="mw-deduplicated-inline-style" href="mw	0							
18	16	<caption id=x tabindex=1 ondeactivate=alert(1)></caption><	1							
19	17	<style onmouseleave="alert(1)">test</style>	1							
20	18	<svg><meta onload=alert(1)></meta>	1							
21	19	<a href="/wiki/Programming_language" title="Programr	0							
22	20	<figcaption onpointerleave=alert(1)>XSS</figcaption>	1							
23	21	 <cite class="citation n	0							

Hình 2.15 File data.csv đưa vào chạy máy học

Bước 1: Tạo file data.csv (dữ liệu file data.csv có sẵn ở trên)

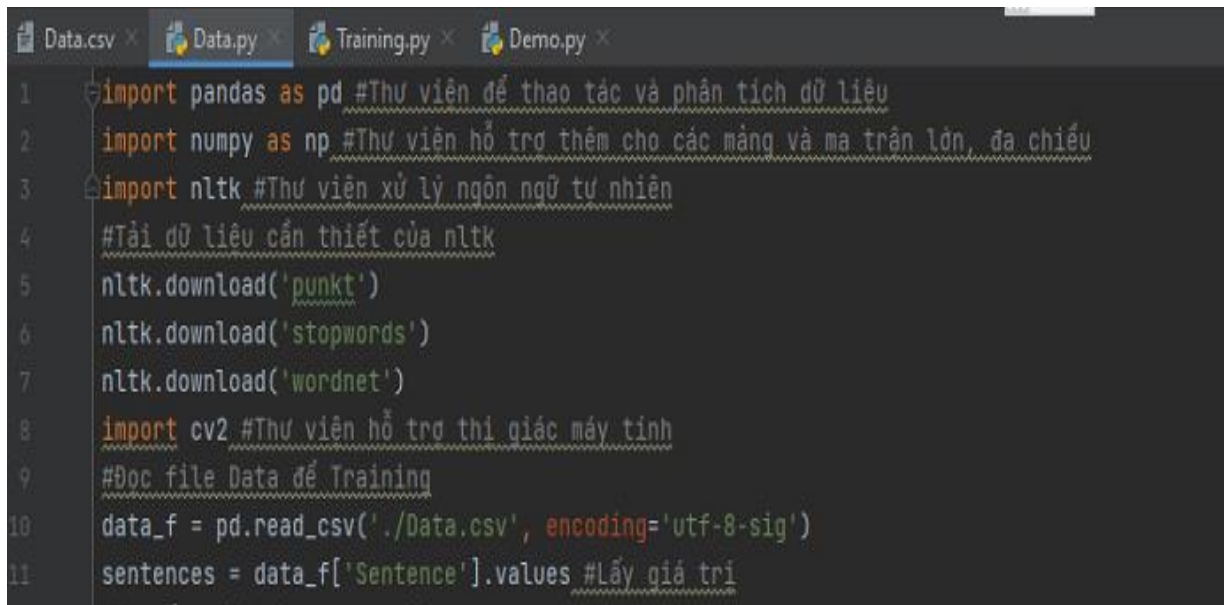


```
1 |Sentence,Label
2 |0,"<li><a href="/wiki/File:Socrates.png" class="image">test</tt>",1
4 |2," </span> <span class="reference-text">Steering for the 1995 ""<a href="/wiki/History_of_autonomous_cars#1990s" class="mw-redirect" title=
5 |3," </span> <span class="reference-text"><cite class="citation web"><a rel="nofollow" class="external text" href="https://www.mileseduca
6 |4," </span> <a href="/wiki/Digital_object_identifier" title="Digital object identifier">doi </a><a rel="nofollow" class="external text"
7 |5,"<li id="cite_note-118"><span class="mw-cite-backlink"><b><a href="#cite_ref-118">^ </a> </b> ",0
8 |6,"<li><a href="/wiki/Contextualism" title="Contextualism">Contextualism </a> </li>",0
9 |7,"<li id="cite_note-Representing_causation-95"><span class="mw-cite-backlink"> <a href="#cite_ref-Representing_causation_95-0"><sup><i><b>
10 |8," <tr><td class="plainlist" style="padding:0 0.1em 0.4em">,0
11 |9, </span>,0
12 |10,"<li><a href="/wiki/Mind%E2%80%93body_problem" title="Mind-body problem">Mind-body problem </a>",0
13 |11,<a onblur=alert(1) tabindex=1 id=x></a><input autofocus>,1
14 |12,"<col draggable="true" ondragenter="alert(1)">test</col>",1
15 |13,<caption onpointerdown=alert(1)>XSS</caption>,1
16 |14, </span>,0
17 |15," </span><link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:r935243608"/> </li>",0
18 |16,<caption id=x tabindex=1 ondeactivate=alert(1)></caption><input id=y autofocus>,1
19 |17,"<style onmouseleave="alert(1)">test</style>,1
20 |18,<svg><meta onload=alert(1)></meta>,1
21 |19,"<li><a href="/wiki/Programming_language" title="Programming language">Programming language </a> </li>",0
22 |20,<figcaption onpointerleave=alert(1)>XSS</figcaption>,1
23 |21," </span> <span class="reference-text"><cite class="citation news"><a rel="nofollow" class="external text" href="https://www.econor
24 |22></div>,0
```

Hình 2.16 Tạo File Data.csv

Bước 2: Tạo file data.py để xử lý dữ liệu của file data.csv vừa tạo

Import các thư viện cần thiết và tiến hành đọc file data.csv



```
1 |import pandas as pd #Thư viện để thao tác và phân tích dữ liệu
2 |import numpy as np #Thư viện hỗ trợ thêm cho các mảng và ma trận lớn, đa chiều
3 |import nltk #Thư viện xử lý ngôn ngữ tự nhiên
4 |#Tải dữ liệu cần thiết của nltk
5 |nltk.download('punkt')
6 |nltk.download('stopwords')
7 |nltk.download('wordnet')
8 |import cv2 #Thư viện hỗ trợ thị giác máy tính
9 |#Đọc file Data để Training
10 |data_f = pd.read_csv('./Data.csv', encoding='utf-8-sig')
11 |sentences = data_f['Sentence'].values #Lấy giá trị
```

Hình 2.17 Import thư viện và tiến hành đọc file Data.csv

Chuyển các giá trị đọc được sang ngôn ngữ mà máy có thể hiểu


```
Data.csv x Data.py x Training.py x Demo.py x
13 def convert_to_ascii(sentence):
14     sentence_ascii = []
15     for i in sentence:
16         if (ord(i) < 8222):
17             if (ord(i) == 8217): # đơn giản hóa ký tự phức tạp
18                 sentence_ascii.append(134)
19             if (ord(i) == 8221):
20                 sentence_ascii.append(129)
21             if (ord(i) == 8220):
22                 sentence_ascii.append(130)
23             if (ord(i) == 8216):
24                 sentence_ascii.append(131)
25             if (ord(i) == 8217):
26                 sentence_ascii.append(132)
27             if (ord(i) == 8211):
28                 sentence_ascii.append(133)
29             if (ord(i) <= 128):
30                 sentence_ascii.append(ord(i))
31             else:
32                 pass
33     zer = np.zeros((10000))
34     for i in range(len(sentence_ascii)):
35         zer[i] = sentence_ascii[i]
36     zer.shape = (100, 100)
```

Hình 2.18 Chuyển các giá trị đọc được sang dạng mã ASCII

Lấy các giá trị sau khi chuyển

```
39 array = np.zeros((len(sentences), 100, 100))
40 #Lấy các giá trị sau khi chuyển
41 for i in range(len(sentences)):
42     image = convert_to_ascii(sentences[i])
43     x = np.asarray(image, dtype='float')
44     image = cv2.resize(x, dsize=(100, 100), interpolation=cv2.INTER_CUBIC) #Phân nhỏ kích thước dữ liệu
45     image /= 128
46     array[i] = image
47     print("Hoàn thành xử lý !!!")
48     data = array.reshape(array.shape[0], 100, 100, 1)
49
50
```

Hình 2.19 Lấy các giá trị sau khi chuyển

Bước 3: Tạo file Training.py dựa vào file data.csv để chạy máy học, sử dụng thư viện tensorflow để hỗ trợ viết hàm

Tạo các đường dẫn thư mục lưu quá trình training, đường dẫn lưu kết quả sau khi training , hàm đánh giá hiệu suất với chia tập data thành hai tập huấn luyện và thử nghiệm với tỉ lệ thử nghiệm là 0.2

```
1  #import các thư viện học máy cần thiết
2  import ...
12 #tạo đường dẫn thư mục lưu quá trình train.
13 NAME = "XSS-cnn-64x2-{}".format(int(time.time()))
14 #gọi hàm TensorBoard từ tensorflow, sẽ có 2 tập dc lưu trong folder là train và validation
15 tensor = TensorBoard(log_dir='./logs/{}'.format(NAME))
16 #tạo các đường dẫn để lưu kết quả sau khi train xong
17 checkpoint_path = './ModelCheckpoint/epoch-{epoch:02d}-val_acc-{val_accuracy:.4f}.h5'
18 model_save = './model/XSS-detection-final.h5'
19 y = data_f['Label'].values #Gán nhãn cho giá trị
20 #sử dụng hàm train_test_split để chia tập data thành tập train và test với số lượng test là 0.2
21 trainX, testX, trainY, testY = train_test_split(data, y, test_size=0.2, random_state=42)
22
```

Hình 2.20 Tạo các đường dẫn thư mục lưu quá trình training, đường dẫn lưu kết quả sau khi training

Tạo mô hình mạng nơ-ron tích chập cơ bản với 11 layer, trong đó có 3 layer là Convolution layer

```
23 #tạo đầu mạng nơ-ron với 7 layer
24 model = Sequential([
25     Conv2D(64, (3, 3), activation=tf.nn.relu, input_shape=(100, 100, 1)),
26     MaxPooling2D(2, 2),
27     Conv2D(128, (3, 3), activation='relu'),
28     MaxPooling2D(2, 2),
29     Conv2D(256, (3, 3), activation='relu'),
30     MaxPooling2D(2, 2),
31     Flatten(),
32     Dense(256, activation='relu'),
33     Dense(128, activation='relu'),
34     Dense(64, activation='relu'),
35     Dense(1, activation='sigmoid')
36 ])
37
```

Hình 2.21 Tạo mô hình mạng nơ-ron tích chập cơ bản với 11 layer

Phân nhỏ kích thước dữ liệu và số lần học máy

Ở đây `batch_size` là kích thước dữ liệu cho 1 lượt học máy, vì dữ liệu đầu vào có thể dài ngắn không xác định rõ, ta cần cố định chúng. Với `num_epoch` thì đây là số lần máy học dữ liệu nhằm tăng chuẩn xác.

```
52  #lấy batchsize và epoch
53  batch_size = 64 #Kích thước phân nhỏ dữ liệu
54  num_epoch = 150 #Số lần máy học
55  # model training
56  #model fitting với số lượng batchsize và epoch đã đặt
57  H = model.fit(trainX, trainY,
58               batch_size=batch_size,
59               epochs=num_epoch,
60               verbose=1,
61               validation_data=(testX, testY),
62               callbacks=[checkpoint, tensor]
63  )
```

Hình 2.22 Phân nhỏ kích thước dữ liệu và số lần học máy

Thực hiện dự đoán model đã tạo, xét tính an toàn dữ liệu

```

64 pred = model.predict(testX)
65 for i in range(len(pred)):
66     if pred[i] > 0.5:
67         pred[i] = 1
68     elif pred[i] <= 0.5:
69         pred[i] = 0
70     true = 0
71     false = 0
72 for i in range(len(pred)):
73     if pred[i] == testY[i]:
74         true += 1
75     else:
76         false += 1
77 # in ra số câu đúng và sai
78 print("Dự đoán ĐÚNG :: ", true)
79 print("Dự đoán SAI :: ", false)
80 attack = 0
81 benign = 0
82 # in ra số lượng data dùng để test
83 for i in range(len(testY)):
84     if testY[i] == 1:
85         attack += 1
86     else:
87         benign += 1
88 print("Tấn công dữ liệu trong tập thử nghiệm :: ", attack)
89 print("Dữ liệu an toàn trong tập thử nghiệm :: ", benign)

```

Hình 2.23 Thực hiện dự đoán model đã tạo, xét tính an toàn dữ liệu

Tạo các hàm đánh giá mô hình

```

91 #tạo các hàm đánh giá mô hình
92 def accuracy_function(tp, tn, fp, fn):
93     accuracy = (tp + tn) / (tp + tn + fp + fn)
94     return accuracy
95
96 def precision_function(tp, fp):
97     precision = tp / (tp + fp)
98     return precision
99
100 def recall_function(tp, fn):
101     recall = tp / (tp + fn)
102     return recall
103

```

```

104 def confusion_matrix(truth, predicted):
105     true_positive = 0
106     true_negative = 0
107     false_positive = 0
108     false_negative = 0
109     for true, pred in zip(truth, predicted):
110         if true == 1:
111             if pred == true:
112                 true_positive += 1
113             elif pred != true:
114                 false_negative += 1
115         elif true == 0:
116             if pred == true:
117                 true_negative += 1
118             elif pred != true:
119                 false_positive += 1
120     accuracy = accuracy_function(true_positive, true_negative, false_positive, false_negative)
121     precision = precision_function(true_positive, false_positive)
122     recall = recall_function(true_positive, false_negative)
123     return (accuracy,
124             precision,
125             recall)

```

Hình 2.24 Tạo các hàm đánh giá mô hình

In ra kết quả đánh giá

```

127 #in ra kết quả đánh giá
128 accuracy, precision, recall = confusion_matrix(testY, pred)
129 print("CNN: \n Chính xác : {0} \n Mức độ chính xác : {1} \n Gọi lại : {2} \n".format(accuracy, precision, recall))
130 print("Hoàn thành quá trình Training !!!")
131 model.save(model_save)

```

Hình 2.25 In ra kết quả đánh giá

Bước 4: Tạo file Demo.py để chạy kết quả training máy học CNN

Import các thư viện cần thiết


```

1 from tensorflow.keras.models import load_model #tải các model cần thiết
2 import cv2 #Cung cấp các mã xử lý phục vụ quy trình thi giác máy tính và Learning Machine.
3 import numpy as np #Thư viện toán học, làm việc với ma trận và mảng với tốc độ xử lý cao.
4 #dùng hàm load_model trong tensorflow để đưa model dạng h5 vào
5 model = load_model('./model/XSS-detection.h5')
6 from Data import convert_to_ascii #import hàm convert_to_ascii từ Data
7

```

Hình 2.26 Import các thư viện cần thiết

Tạo hàm dự đoán XSS với số lần dự đoán

```

8 #tạo hàm dự đoán XSS
9 def predict_cross_site_script():
10     repeat = True
11     beautify = ''
12     for i in range(20): #Lặp 20 lần dự đoán
13         beautify += "="

```

Hình 2.27 Tạo hàm dự đoán XSS với số lần dự đoán

Tạo đầu nhập input (số hoặc chữ) : nếu giá trị nhập vào âm thì trả lại giá trị False, ngược lại sẽ tiếp tục thực hiện quá trình tiếp theo

```

14 #tạo đầu nhập input là giá trị số hoặc chữ
15 print(beautify)
16 input_val = input("Giá trị cần kiểm tra: ") #Nhập vào giá trị cần kiểm tra.
17 print(beautify)
18 # nếu đầu vào - thì trả lại giá trị False
19 if input_val == '0':
20     repeat = False
21 #nếu đúng thì code sẽ gọi hàm convert qua dạng mã ascii cho đầu vào input từ đó tách các layer phân tích
22 # cho đầu mạng nơ-ron của chúng ta nhận diện
23 if repeat == True: #Nếu giá trị trả về là True
24     image = convert_to_ascii(input_val) #Chuyển giá trị đầu vào sang ASCII
25     x = np.asarray(image, dtype='float')
26     # "a" - dữ liệu đầu vào
27     # "dtype" - kiểu dữ liệu được suy ra từ dữ liệu đầu vào
28     image = cv2.resize(x, dsize=(100, 100), interpolation=cv2.INTER_CUBIC)
29     # "x" - nguồn đưa vào
30     # "dsize" - kích thước mong muốn cho đầu ra
31     # "interpolation=cv2.INTER_CUBIC" - Cho phép nội suy 2 chiều trên vùng lân cận 4x4 pixel
32     image.shape = (1, 100, 100, 1) #Gán kích thước cho image
33     image /= 128 #Ảnh với kích thước 128 pixel
34     prediction = model.predict(image) #Dự đoán dự vào model

```

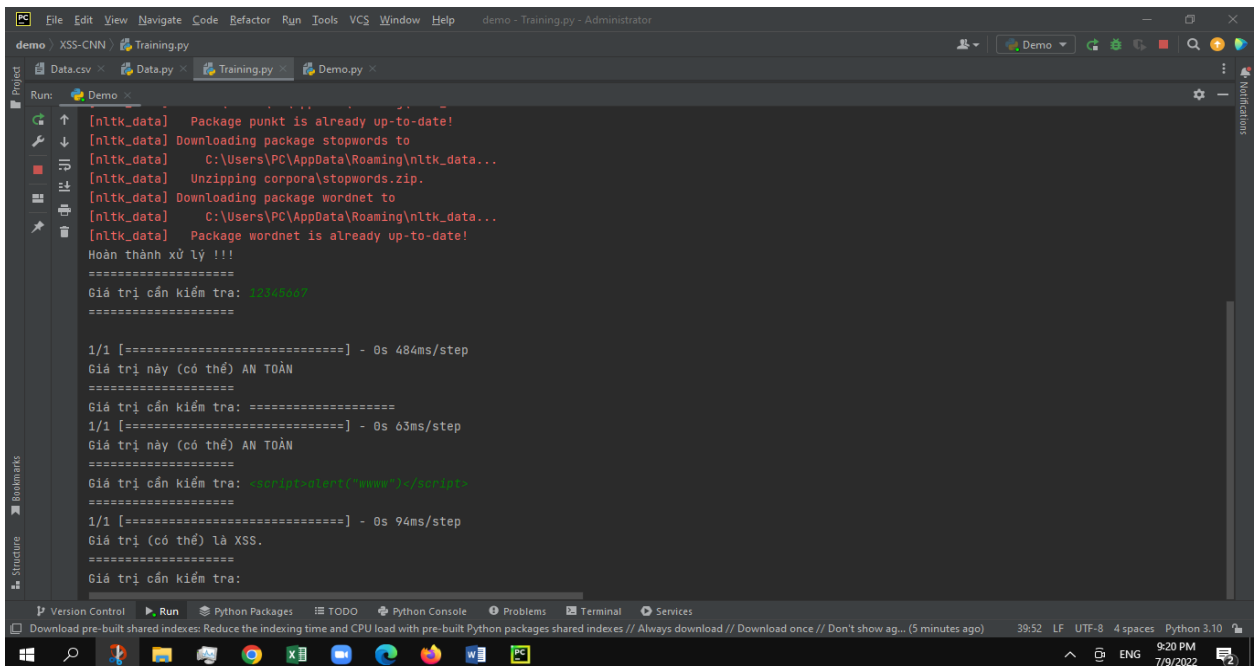
Hình 2.28 Tạo đầu nhập input

Kiểm tra kết quả thu được : kết quả cho ra có độ tin cậy lớn hơn 50% sẽ là XSS, ngược lại là giá trị an toàn

```
35 #nếu như kết quả có độ tin cậy lớn hơn 50% thì sẽ XSS
36 if prediction > 0.5:
37     print("Giá trị (có thể) là XSS.")
38     # còn nếu không thì an toàn
39 else:
40     print("Giá trị này (có thể) AN TOÀN")
41     #đóng gói hàm
42     predict_cross_site_script()
43 elif repeat == False:
44     print(" Good Bye ")
45 #đóng gói toàn bộ code
46 if __name__ == '__main__':
47     predict_cross_site_script()
48
```

Hình 2.29 Kiểm tra kết quả thu được sau khi training

Kết quả thu được khi chạy file thực nghiệm

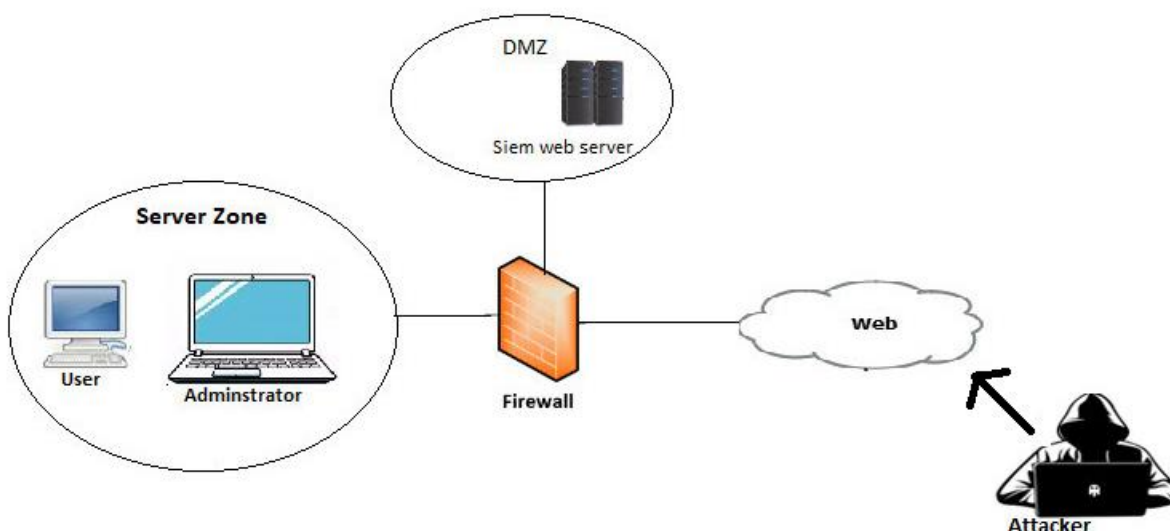


```
demo - Training.py - Administrator
demo - XSS-CNN - Training.py
Data.csv - Data.py - Training.py - Demo.py
Run: Demo
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\PC\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\stopwords.zip.
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\PC\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
Hoàn thành xử lý !!!
=====
Giá trị cần kiểm tra: 1234567
=====
1/1 [=====] - 0s 484ms/step
Giá trị này (có thể) AN TOÀN
=====
Giá trị cần kiểm tra: =====
1/1 [=====] - 0s 63ms/step
Giá trị này (có thể) AN TOÀN
=====
Giá trị cần kiểm tra: <script-alert(1/www(1)/script>
=====
1/1 [=====] - 0s 94ms/step
Giá trị (có thể) là XSS.
=====
Giá trị cần kiểm tra:
```

Hình 2.30 Kết quả thu được khi chạy file thực nghiệm

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG PHÁT HIỆN TẤN CÔNG MẠNG DỰA TRÊN LƯU LƯỢNG BẤT THƯỜNG TRONG HỆ THỐNG MẠNG

3.1 Mô hình triển khai



Hình 3.1 Mô hình triển khai hệ thống phát hiện tấn công mạng dựa trên lưu lượng bất thường trong hệ thống mạng

3.1.1. Thông tin thiết bị triển khai

STT	Tên		Interface (VMnet1)	Cấu hình
1	Webserver_S plunk(cài siem lun)	IP	192.168.2.15	OS: FreeBSD Ram: 1GB
		SM	255.255.255.0	
		DG		
		DNS		
2	Adminstrator	IP	192.168.2.19	OS: windowXp Ram: 2.5GB
		SM	255.0.0.0	

3	Attacker	DG		OS: Windows 10 Ram: 12GB
		DNS	8.8.8.8	
		IP	192.168.2.12	
		SM	255.255.255.0	
		DG	192.168.2.253	
		DNS	8.8.8.8	

Bảng 3.1 Thông tin thiết bị triển khai.

3.2. Thực nghiệm kịch bản tấn công

3.2.1. Tấn công SQL Injection vào Web Server_Splunk

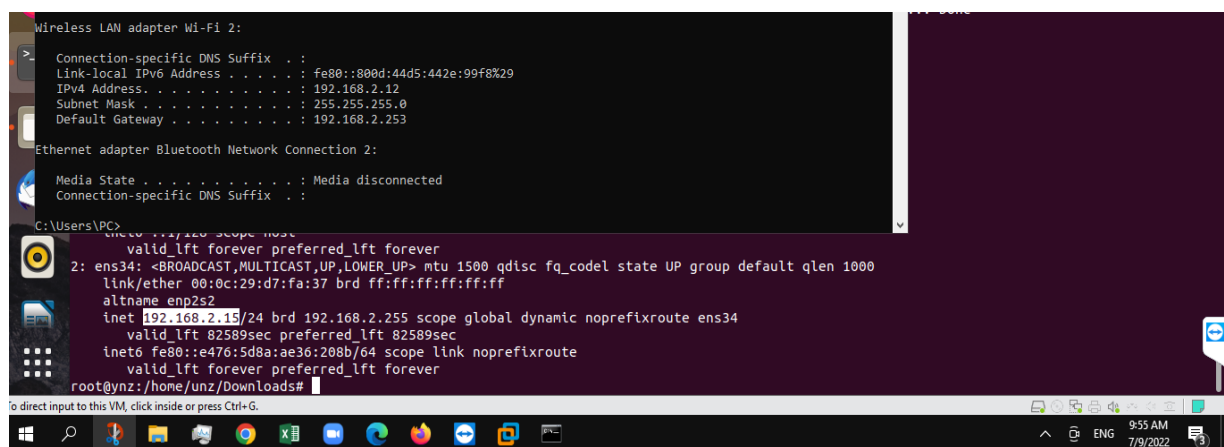
3.3.1.1 Mục đích

Phát hiện tấn công mạng dựa trên lưu lượng bất thường trên hệ thống siem lunx

Mô tả: Máy Attacker tấn công SQL Injection vào Web Server

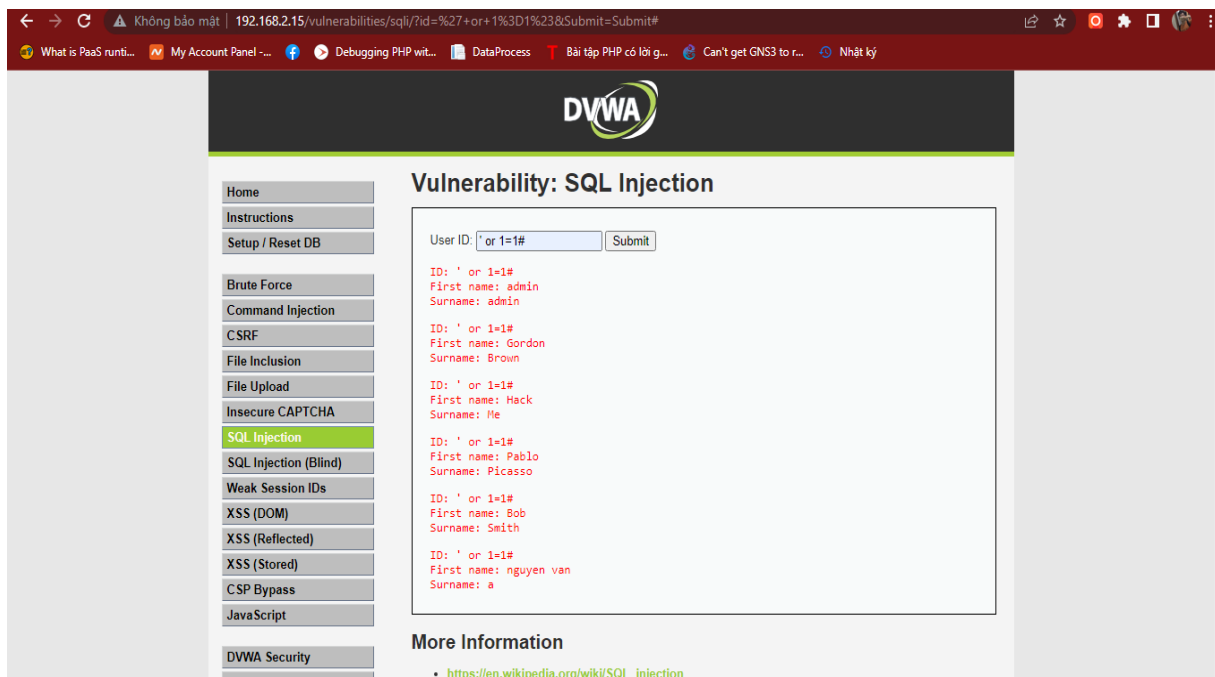
Tiến hành inject mã SQL query/command “*UNION SELECT user, password FROM users #*” vào input của websites đang được chạy trên Web Server.

IP của máy tấn công:



Hình 3.2 IP của máy tấn công

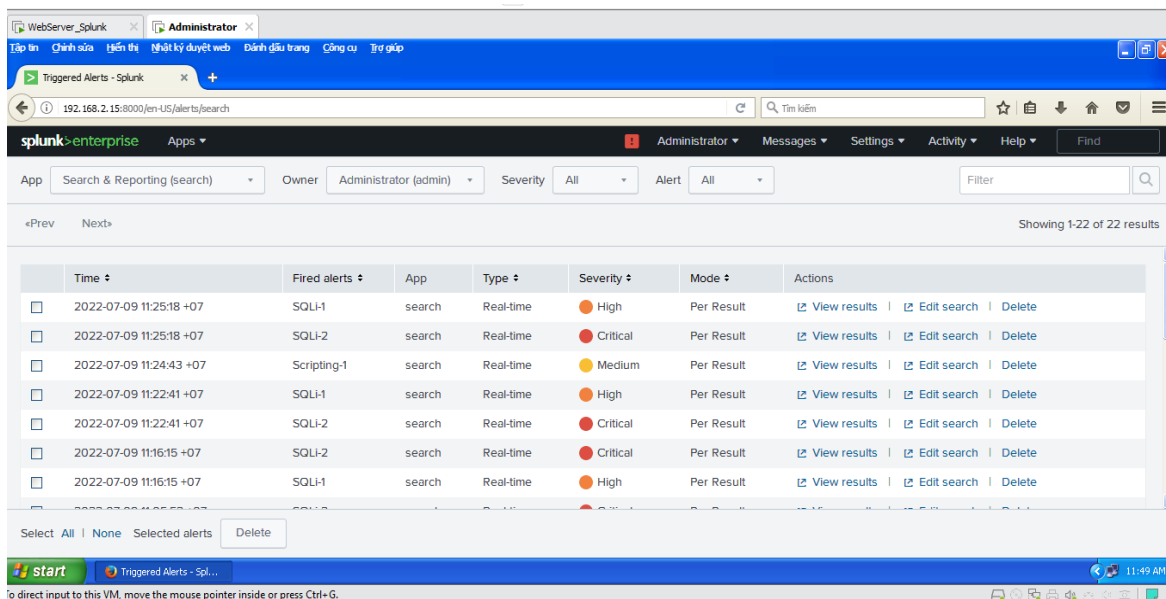
3.3.1.2 Quá trình tấn công



Hình 3.3 Quá trình tấn công SQL Injection vào Web Server

3.3.1.3 Kết quả

Log được ghi lại bởi SIEM Splunk



Hình 3.4 Kết quả ghi log của SIEM Splunk

3.3.1.4 Đánh giá kịch bản

Đã phát hiện được cuộc tấn công SQL Injection.

3.3.2 Tấn công XSS vào Web Server

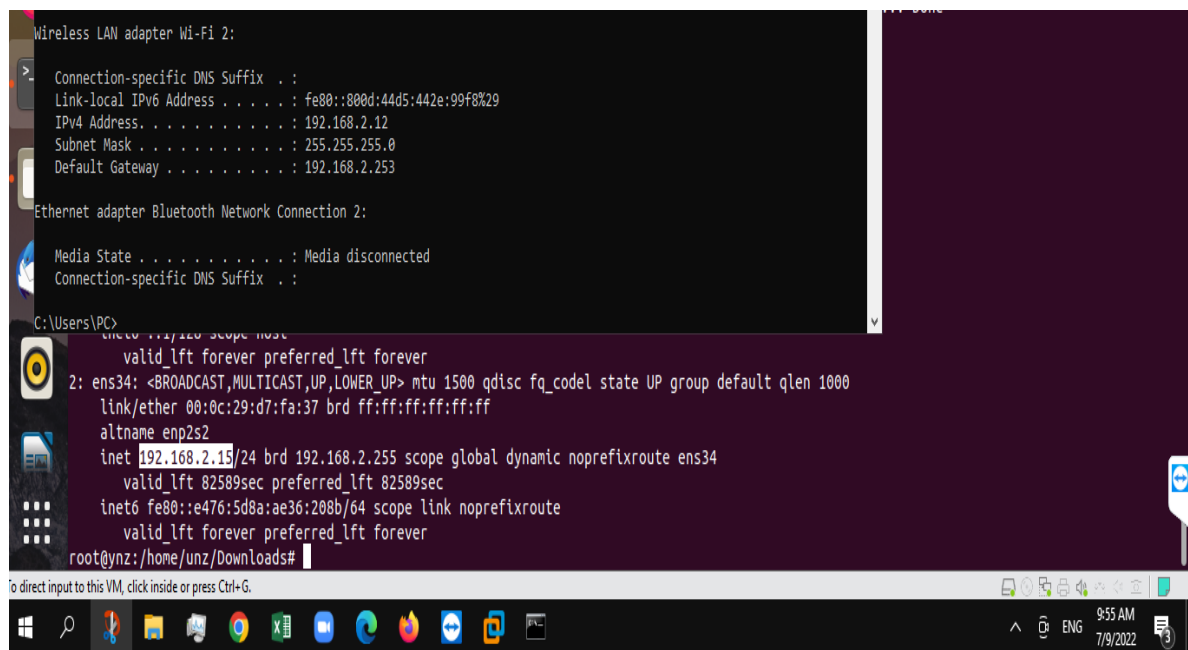
3.3.2.1 Mục đích

Phát hiện tấn công mạng dựa trên lưu lượng bất thường trên hệ thống siem lunk

Mô tả:Máy Attacker tấn công XSS vào Web Server:

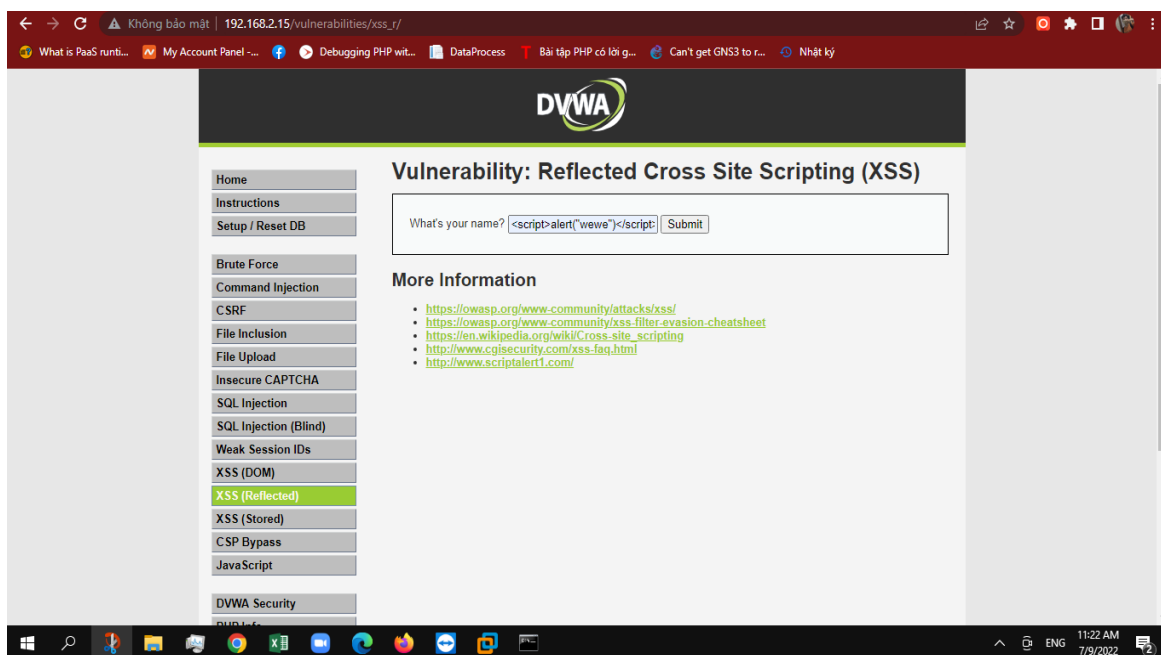
Tiến hành chèn mã độc “*<script>alert(document.cookie)</script>*” (thông qua đoạn Script) để thực thi chúng trên Web Server.

IP máy tấn công:



Hình 3.5 IP của máy tấn công

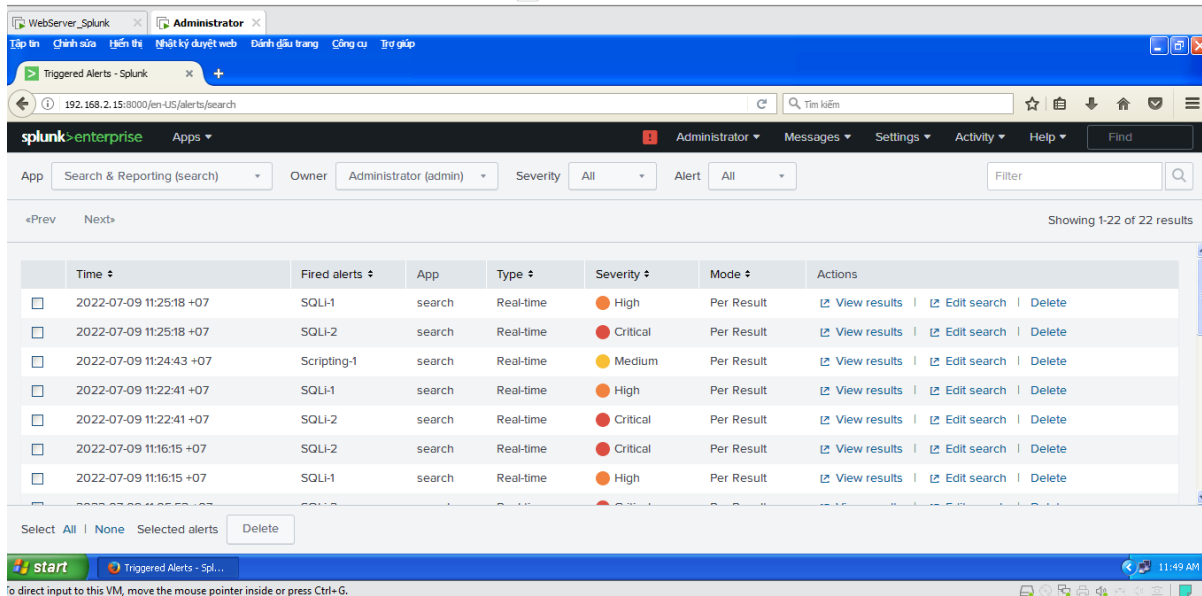
3.3.2.2 Quá trình tấn công



Hình 3.6 Quá trình tấn công XSS vào Web Server

3.3.2.3 Kết quả

Log được ghi lại bởi SIEM Splunk



Hình 3.7 Kết quả bắt log của SIEM Splunk

3.3.2.4 Đánh giá kịch bản

Đã Phát hiện được cuộc tấn công XSS.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau khi hoàn thành Khóa luận tốt nghiệp, trải qua một thời gian không ngắn nhưng cũng không quá dài, nhóm chúng em cũng đạt được những kết quả cho bản thân mình. Tìm hiểu về Hệ thống giám sát an ninh sự kiện SIEM, chúng em hiểu được tầm quan trọng của hệ thống, các thành phần, lợi ích cũng như các giải pháp triển khai hệ thống SIEM. Từ đó ứng dụng vào triển khai hệ thống phát hiện các lưu lượng bất thường trên hệ thống mạng. Bên cạnh đó, chúng em nghiên cứu các thuật toán máy học như: SVM, LSTM, RNN, CNNs,... Đặc biệt là qua quá trình thực nghiệm thuật toán máy học CNNs, chúng em hiểu được cơ chế hoạt động cũng như lợi ích của nó trong việc dự đoán bất thường. Và cuối cùng, chúng em đã triển khai được một hệ thống phát hiện các lưu lượng mạng bất thường trên hệ thống mạng dựa vào các dự đoán bất thường và hệ thống hỗ trợ phát hiện bất thường SIEM.

Ngoài những kết quả đã đạt được, và thời gian hoàn thành khóa luận có hạn nên chúng em còn những hạn chế nhất định. Chúng em vẫn chưa thực nghiệm hết các thuật toán máy học ở trên. Ngoài ra, Hệ thống SIEM phát hiện bất thường và ghi log nhưng chỉ khi nào vào hệ thống mới xem được cảnh báo. Điều này vẫn chưa phải là tối ưu nhất cho hệ thống.

Hướng phát triển của nhóm em nếu có thêm thời gian hoàn thành khóa luận đó là khắc phục các hạn chế chưa đạt được. Chạy thử nghiệm được hết các thuật toán máy học: SVM, LSTM, RNN để hiểu rõ hơn về cơ chế hoạt động của chúng cũng hiểu hơn về lợi ích và ứng dụng của chúng. Còn về phần hệ thống phát hiện các lưu lượng bất thường trên hệ thống mạng, chúng em sẽ triển khai thêm cấu hình dịch vụ gửi cảnh báo về mail cho admin, lúc đó admin không cần trực tại server mà vẫn có thể nhận được cảnh báo tấn công để kịp thời xử lý.

TÀI LIỆU THAM KHẢO

- [1] Miedaner, T. (July 26, 2018). Open Source SIM - Generation 2 Graylog Appliance. (Reality Check Series Book 6) Kindle Edition
- [2] Miedaner, T. (February 4, 2017). Security Incident Detection and Response (Reality Check Series Book 5) . Kindle
- [3] Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. ACM Conference on Computer and Communications Security (CCS), 2017.
- [4] A. Shenfield, D. Day and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," ICT Express, vol. 4, no. 2, pp. 95-99, 2018.
- [5] Mengying Wang ; Lele Xu ; Lili Guo, "Anomaly Detection of System Logs Based on Natural Language Processing and Deep Learning" 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), IEEE, 29 November 2018.
- [6] Akinsola, Jide E. T, Awodele, Oludele, Idowu, Sunday A, SQL Injection Attacks Predictive Analytics Using Supervised Machine Learning Techniques, International Journal of Computer Applications Technology and Research Volume 9– Issue 04, 139-149, ISSN:-2319–8656, 2020

LINK THAM KHẢO

- [7] <https://www.iworld.com.vn/quan-ly-su-kien-va-thong-tin-bao-mat-siem-la-gi/>
- [8] <https://vietnetco.vn/solutions/soc-and-noc/siem>
- [9] <https://ntshanoi.com.vn/giai-phap-siem.html#:~:text=SIEM%20l%C3%A0%20vi%E1%BA%B Ft%20t%E1%BA%AFt%20c%E1%BB%A7a,t%C3%A2n%20ti%E1%BA%BFn%20nh%E1%BA%A5t%20hi%E1%BB%87n%20nay.>
- [10] <https://viblo.asia/p/phat-hien-xam-nhap-mang-bang-phat-hien-bat-thuong-dua-tren-phuong-phap-svm-phan-3-gioi-thieu-may-ho-tro-vecto-svm-gDVK2RQvKLj>
- [11] https://phamdinhhkhanh.github.io/2019/04/22/Ly_thuyet_ve_mang_LSTM.html

- [12] <https://websitehcm.com/recurrent-neural-network-rnn-trong-tensorflow/>
- [13] <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>
- [14] <https://topdev.vn/blog/hadoop-la-gi/>
- [15] <https://phambinh.net/bai-viet/tim-hieu-ve-hadoop-hdfs/>
- [16] <https://www.datashieldprotect.com/tools/splunk>