HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Lê Ngọc Khoa

NGHIÊN CỨU PHƯƠNG PHÁP PHÁT HIỆN TẤN CÔNG ỨNG DỤNG WEB SỬ DỤNG HỌC SÂU CNN

Chuyên ngành: Khoa học máy tính
Mã số: 8.48.01.01

Hà Nội 2024

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: PGS.TS. HOÀNG XUÂN DẬU

Phản biện 1: ……

Phản biện 2: ……

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .....năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Ứng dụng web hay web application, web app là một trình ứng dụng mà có thể tiếp cận qua web thông qua mạng như Internet hay intranet. Web Application thường được lưu trữ trên một máy chủ từ xa và người dùng có thể truy cập nó thông qua việc sử dụng Phần mềm được gọi là trình duyệt web. Các Web Application có thể được thiết kế cho nhiều mục đích sử dụng khác nhau và có thể được sử dụng bởi bất kỳ ai, một tổ chức hoặc một cá nhân.

Trên nền tảng internet, các ứng dụng web đang chiếm tỷ lệ không hề nhỏ. Các gã khổng lồ công nghệ (Google, Facebook, Amazon…) đều có những ứng dụng web với số lượng lên tới hàng tỉ người dùng. Chính vì vậy mà yếu tố bảo mật và phòng chống tấn công ứng dụng web trở nên quan trọng hơn bao giờ hết.

Tấn công ứng dụng web có thể gây ra nhiều hậu quả nghiêm trọng đối với cả người dùng, tổ chức và các dự án trực tuyến. Dưới đây là một số hậu quả quan trọng của tấn công ứng dụng web:

- Mất thông tin cá nhân
- Tiết lộ thông tin doanh nghiệp
- Tấn công từ chối dịch vụ (Denial of Service - DoS)
- Xâm nhập trái phép hệ thống
- Mất lợi nhuận
- Sự can thiệp pháp lý
- Phát triển mã độc

Do các cuộc tấn công web có thể dẫn đến những hậu quả nặng nề cho các cá nhân, tổ chức nên việc nghiên cứu phương pháp hiệu quả cho phát hiện tấn công ứng dụng web là rất cần thiết. Mặc dù đã có một số đề xuất và công cụ phát hiện tấn công ứng dụng web, việc nghiên cứu nhằm nâng cao độ chính xác phát hiện, cảnh báo, ngăn chặn sai vẫn cần được tiếp tục triển khai. Hơn nữa các phương pháp học máy và đặc biệt là học sâu đã và đang được ứng dụng hiệu quả trong giải quyết các bài toán thực tiễn của lĩnh vực khoa học máy tính cũng như an toàn thông tin. Đây cũng

là mục tiêu của đề án này, với đề tài "Nghiên cứu phương pháp phát hiện tấn công Web dựa trên học sâu".

2. Tổng quan vấn đề cần nghiên cứu

Hiện nay, tình trạng tấn công ứng dụng web đang xảy ra ngày càng nhiều, đặc biệt là đối với các cơ sở trọng yếu, khối ngân hàng, tài chính và các doanh nghiệp lớn. Thách thức đó đặt ra là cần phải thực hiện các phương pháp bao mật ứng dụng web. Để bảo vệ ứng dụng web khỏi các cuộc tấn công, cần phải triển khai một loạt các biện pháp phát hiện tấn công web hiệu quả. Hiện tại có nhiều phương pháp, giải pháp được ứng dụng để phát hiện các cuộc tấn công web như hệ thống phát hiện xâm nhập (IDS), phát hiện dựa trên chữ ký, dựa trên hành vi, sử dụng tưởng lửa ứng dụng web (WAF), phân tích dữ liệu lưu lượng mạng (theo dõi bất thường, chủ động phòng chống DOS), sử dụng các tập rule (xây dựng dựa trên các mẫu tấn công đã biết), giám sát hệ thống và dựa trên các kỹ thuật học máy, học sâu.

Gần đây, các giải pháp phát hiện tấn công ứng dụng web sử dụng các kỹ thuật thống kê, học máy, học sâu được triển khai và áp dụng tương đối rộng rãi, cho kết quả khả quan. Theo đó các kỹ thuật thống kê, học máy, học sâu được sử dụng để xây dựng mô hình phân loại các loại tấn công từ tập dữ liệu gồm các request bình thường các các request chứa các mã khai thác đã biết. Nhờ số lượng request thu thập lớn và phương pháp xử lý hiệu quả, việc phát hiện các cuộc tấn công ứng dụng web dựa trên ký thuật thống kê, học máy, học sau cho độ chính xác cao và tỷ lệ cảnh báo sai thấp.

Theo hướng sử dụng học máy học xâu để xây dựng mô hình phát hiện tấn công có thể liệt kê một số đề xuất, như nhóm tác giả Tikam Alma và Manik Lal Das đề xuất phương pháp phát hiện hiện tấn công ứng dụng web sử dụng học sâu trên cơ sở mạng nơ ron LSTM (Long Short Term Memory) và cung cấp đầu vào theo thứ tự [3]. Kết quả thử nghiệm cho thấy độ chính xác rất cao với tỉ lệ 0.9968. Prasanna Kottapalle đề xuất mô hình kết hợp CNN-LSTM cho IDS trên tập dữ liệu KDD99 cũng đạt được độ chính xác cao (99.78%), một số thử nghiệm với các mô hình khác như SVM (98.20%), DBN (98.59%) CNN* (99.23%) [4]. Ngoài ra tác giả Abdu Salam cùng cộng sự đề xuất mô hình phát hiện tấn công ứng dụng web sử dụng học sâu trên cơ

sở mạng nơ ron CNN, kết quả đạt được cũng rất tích cực với tỉ lệ chính xác 94% và tỉ lệ phân loại các lỗ hổng cũng ở mức cao (DDOS – 91%, SQL Injection – 90%, XSS – 92%) [5].

Đề án này đề xuất sử dụng các phương pháp học sâu, gồm CNN để xây dựng mô hình phát hiện tấn công ứng dụng web. Ưu điểm của các phương pháp học sâu là giảm thiểu việc trích xuất chọn đặc trưng và khả năng xử lý các tập dữ liệu lớn.

3. Mục đích nghiên cứu

Đề án nghiên cứu, khảo sát các phương pháp phát hiện tấn công ứng dụng web và tập trung cài đặt, thử nghiệm và đánh giá mô hình phát hiện tấn công web dựa trên học sâu.

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các request bình thường và request chứa payload tấn công, các phương pháp học máy và học sâu
- Phạm vi nghiên cứu: Giới hạn các request với hai method GET và POST.

5. Nội dung

Chương 1: Các phương pháp tấn công ứng dụng web và cách phòng chống

1.1 Tổng quan về ứng dụng web

- Ứng dụng web là gì?
- Mô hình client – server của ứng dụng web
- Giao thức HTTP/HTTPS

1.2 Các Lỗ hổng, rủi do bảo mật ứng dụng web

- Lỗ hổng bảo mật ứng dụng web là gì?
- Khái niệm unstrusted data, unsafe method.
- Top 10 rủi do/lỗ hổng OWASP
- Một số lỗ hổng bảo mật Web phổ biến

1.3 Các giải pháp bảo vệ và phòng chống tấn công ứng dụng web

- Triển khai các giải pháp bảo mật bảo vệ ứng dụng web
- Cấu hình, cập nhật phiên bản ứng dụng định kỳ

- Không tin tưởng dữ liệu do người dung cung cấp
- Phòng thủ theo chiều sâu

Chương 2: Phát hiện tấn công web sử dụng học sâu

2. 1 Khái quát về học máy và học sâu

- Khái quát về học máy
- Khái quát về học sâu
- Một số phương pháp học sâu

2. 2 Phát hiện tấn công ứng dụng web dựa trên học sâu CNN

- Mô hình phát hiện tấn công ứng dụng web sử dụng học sâu CNN
- Các giai đoạn xử lý
- Tiêu chuẩn đánh giá mô hình

2. 3 Kết chương

Chương 3: Thử nghiệm và đánh giá

3.1 Tập dữ liệu thử nghiệm.

3.2 Tiền xử lý dữ liệu

3.3 Huấn luyện và kiểm tra

- Môi trường thử nghiệm
- Kết quả và nhận xét

3.4 Cài đặt thử nghiệm mô đun phát hiện tấn công ứng dụng web

3.5 Kết chương

Kết luận

CHƯƠNG 1: CÁC PHƯƠNG PHÁP TẤN CÔNG ỨNG DỤNG WEB VÀ CÁCH PHÒNG CHỐNG

1.1 Tổng quan về ứng dụng web

1.1.1 Ứng dụng web là gì?

Ngày nay, với xu hướng phát triển mạnh mẽ của công nghệ số các cơ quan tổ chức, doanh nghiệp tăng cường truyền thông quảng bá hình ảnh trên không gian mạng. Ứng dụng web nổi lên như một phương thức được sử dụng phổ biến giúp cho việc tiếp cận người dùng, khách hàng dễ dàng hơn.

1.1.2 Mô hình client – server của ứng dụng web

Trong mô hình, server được hiểu như web server- máy chủ web. Máy chủ web là thành phần cốt lõi để website hoạt động, bao gồm các thành phần con như cơ sở dữ liệu, các đoạn mã, tập lệnh và một số thành phần khác. Máy chủ web sẽ có vai trò phân tích và xử lý yêu cầu của client, đồng thời phân phối nội dung đến client thông qua các phương thức như HTTP (Hypertext Transfer Protocol) hoặc phương thức truyền file như FTP (File Transfer Protocol).

1.1.3 Giao thức HTTP/HTTPS

HTTP là từ viết tắt của Hyper Text Transfer Protocol nghĩa là Giao thức Truyền tải Siêu Văn Bản hoạt động theo kiểu yêu cầu - phản hồi. Đây là nền tảng của bất kỳ sự trao đổi dữ liệu nào trên các ứng dụng Web và cũng là giao thức được sử dụng trong giao tiếp giữa máy khách (client) và máy chủ (server). Theo đó, máy khách (client) tạo ra một yêu cầu (HTTP request) và gửi nó đến máy chủ HTTP ở cổng biết trước (Well-known port).

1.2 Các lỗ hổng bảo mật trong ứng dụng web

1.2.1 Lỗ hổng bảo mật ứng dụng web là gì?

Lỗ hổng bảo mật ứng dụng web là các điểm yếu bảo mật của một ứng dụng web mà có thể bị tận dụng để đe dọa tính toàn vẹn, quyền riêng tư, hoặc khả năng sẵn sàng của hệ thống. Các lỗ hổng này có thể dẫn đến việc mất thông tin nhạy cảm, thất

bại trong việc duy trì tính khả dụng của dịch vụ, hoặc bị tấn công bởi các tin tặc hoặc kẻ tấn công khác.

1.2.2 Top 10 lỗ hổng, rủi ro theo OWASP

OWASP Top 10 là một báo cáo được cập nhật thường xuyên về các nguy cơ bảo mật đối với bảo mật ứng dụng web, tập trung vào 10 rủi ro/lỗ hổng quan trọng nhất. Báo cáo được tổng hợp bởi một nhóm các chuyên gia bảo mật từ khắp nơi trên thế giới.

1.2.3 Một số lỗ hổng bảo mật Web phổ biến

- SQL injection
- Cross-Site Scripting (XSS)
- Cross-site request forgery
- Path Traversal
- File Inclusion

1.3 Phương pháp phát hiện và phòng chống tấn công ứng dụng web

Bảo mật của ứng dụng web đặt trọng tâm vào ba lớp quan trọng: Bảo mật Mạng, Bảo mật Máy chủ, và Bảo mật Ứng dụng.

Kết luận chương 1

Chương 1 đã trình bày khái quát về ứng dụng web, giao thức HTTP/HTTPS, các rủi ro, lỗ hổng bảo mật trong top 10 OWASP, một số lỗ hổng tấn công ứng dụng web phổ biến. Bên cạnh đó, chương 1 còn nêu các phương pháp, giải pháp và mô hình phòng thủ chiều sâu trong phòng chống tấn công ứng dụng web.

Chương 2 sẽ giới thiệu về học máy, học sâu và phương pháp phát hiện tấn công ứng dụng web dựa trên mô hình học sâu CNN.

CHƯƠNG 2: PHÁT HIỆN TẤN CÔNG WEB SỬ DỤNG HỌC SÂU

2.1. Khái quát về học máy và học sâu

2.1.1. Khái quát về học máy

Học máy là khả năng của chương trình máy tính sử dụng kinh nghiệm, quan sát, hoặc dữ liệu trong quá khứ để cải thiện công việc của mình trong tương lai thay vì chỉ thực hiện theo đúng các quy tắc đã được lập trình sẵn. Chẳng hạn, máy tính có thể học cách dự đoán dựa trên các ví dụ, hay học cách tạo ra các hành vi phù hợp dựa trên quan sát trong quá khứ.

Sử dụng những dạng kinh nghiệm và dạng biểu diễn khác nhau dẫn tới những dạng học máy khác nhau. Có bốn dạng học máy chính như sau:

- Học có giám sát (supervised learning)
- Học không giám sát (Unsupervised learning)
- Học nửa giám sát (Semi supervised learning)
- Học tăng cường (reinforcement learning)

2.1.2 Khái quát về học sâu

Học sâu là một nhánh của học máy hoàn toàn dựa trên mạng nơ ron nhân tạo, vì mạng nơ ron sẽ bắt chước bộ não con người nên học sâu cũng là một loại bắt chước bộ não con người. Nhiều mô hình học sâu được áp dụng trong các lĩnh vực như: Thị giác máy tính, xử lý ngôn ngữ tự nhiên, phát hiện bất thường...học sâu mô hình hóa các mối quan hệ và khái niệm phức tạp bằng cách sử dụng nhiều cấp độ biểu diễn.

Giới thiệu mạng nơ ron

Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một hệ thống tính toán có cấu trúc tương tự như mạng nơ-ron trong não người. Được thiết kế để mô phỏng cách nơ-ron làm việc, ANN là một phần quan trọng của lĩnh vực trí tuệ nhân tạo (AI).

Một ANN bao gồm các "nơ-ron" được tổ chức thành các lớp: lớp đầu vào, lớp ẩn (nếu có), và lớp đầu ra. Mỗi nơ-ron trong lớp được kết nối với tất cả các nơ-ron

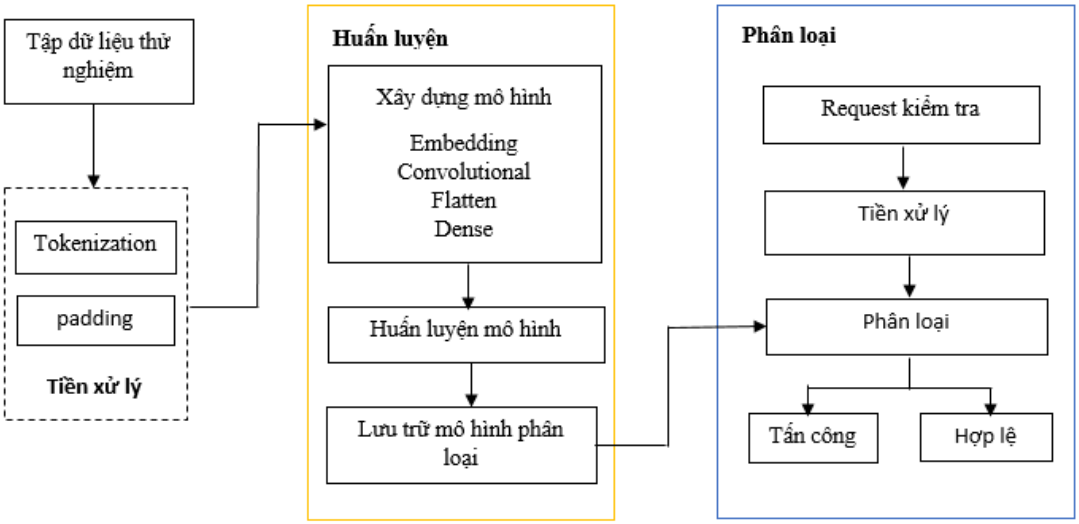
trong lớp liền kề bằng các trọng số. Các trọng số này được điều chỉnh trong quá trình huấn luyện để mô hình có thể học từ dữ liệu.

2.1.3 Một số phương pháp học sâu

- Mạng nơ ron sâu (Deep Neural Network-DNN)
- Mạng nơ ron tích chập (Convolutional Neural Network)

2.2 Khái quát về học máy và học sâu

2.2.1 Giới thiệu mô hình



Mô hình phát hiện tấn công ứng dụng web dựa trên học sâu CNN gồm ba giai đoạn chính: Tiền xử lý dữ liệu, giai đoạn huấn luyện và giai đoạn phân loại. Dữ liệu đầu vào là các HTTP request được kết hợp từ các nguồn khác nhau.

2.2.2 Các giai đoạn xử lý

a. Tiền xử lý:

Trong quá trình tiền xử lý dữ liệu, hai bước quan trọng là "Tokenization" và "Padding". Trong bước "Tokenization", dữ liệu văn bản được chuyển đổi thành chuỗi số duy nhất, tiện lợi cho việc đưa vào mạng nơ-ron. Tiếp theo, trong bước "Padding", các chuỗi số được điều chỉnh độ dài để đồng nhất, giúp

cho mô hình có thể xử lý chúng một cách hiệu quả hơn trong quá trình huấn luyện và dự đoán.

b. Huấn luyện:

Các bước của giai đoạn huấn luyện như sau:

- Lớp Nhúng (Embedding Layer):
  Lớp nhúng sử dụng để biểu diễn các thông tin trong các request dưới dạng các vectơ nhúng có số chiều thấp.
- Lớp Tích Chập (Convolutional Layer):
  Lớp tích chập sẽ quét qua các vectơ nhúng biểu diễn cho các HTTP request để trích xuất các đặc trưng cục bộ.
  Các bộ lọc trong lớp tích chập sẽ học được các mẫu hoặc đặc điểm của các request, giúp mô hình nhận biết các dấu hiệu của các loại tấn công.
- Lớp Phẳng Hóa (Flatten Layer):
  Lớp phẳng hóa sẽ chuyển đổi đầu ra từ lớp tích chập thành một vectơ 1 chiều.
- Lớp Kết Nối Đầy Đủ (Dense Layer):
  Lớp kết nối đầy đủ sẽ nhận đầu vào từ lớp phẳng hóa và thực hiện quá trình phân loại, tức là dự đoán xem một request có chứa tấn công hay không.

c. Phân loại:

- Dữ liệu đầu vào: Là request người dùng gửi lên server.
- Xử lý dữ liệu, vector hóa: Request sẽ được tiền xử lý với tokenization và padding, sau đó được vector hóa để phù hợp với mô hình.
- Dự đoán và trả về kết quả: sau khi được vector hóa, vector này sẽ được dự đoán có là một dạng tấn công hay không thông qua mô hình học sâu CNN. Nếu phát hiện tấn công, mô hình sẽ trả về kết quả là dạng tấn công cụ thể.

2.2.3 Tiêu chuẩn đánh giá mô hình

Để đánh giá được độ chính xác của mô hình ta sử dụng một ma trận được gọi là confusion matrix.

Để đánh giá được độ chính xác của mô hình ta sử dụng một ma trận được gọi là confusion matrix.

		Giá trị thực tế	
Giá trị dự đoán		Positive (1)	Negative (0)
	Positive (1)	TP	FP
	Negative (0)	FN	TN

Hình 2- 1 Confusion matrix đánh giá độ chính xác mô hình học sâu

Các chỉ số trong ma trận gồm có:

- True positive (TP): các request tấn công được phân loại chính xác là các request tấn công
- True negative (TN): các request hợp lệ được phân loại chính xác là các request hợp lệ
- False positive (FP): các request hợp lệ mà được phân loại không chính xác là các request tấn công
- False negative (FN): Các request tấn công được phân loại không chính xác là các request hợp lệ.

Confusion matrix có dạng bảng, trong đó hàng của ma trận thể hiện các lớp thực tế, còn cột thể hiện các lớp được dự đoán bởi mô hình.

Thông qua confusion matrix, chúng ta có thể tính toán các chỉ số đánh giá hiệu suất như độ chính xác (accuracy), độ chính xác của từng lớp (precision), độ phủ

(recall), F1-score, và nhiều metric khác để đánh giá hiệu suất của một mô hình phân loại.

Kết luận chương 2

Chương hai đã trình bày các khái niệm cơ bản về học máy, học sâu và mô hình phát hiện tấn công ứng dụng web dựa trên học sâu. Các lý thuyết và chỉ số đánh giá hiệu quả sẽ được vận dụng để xây dựng và đánh giá mô hình phát hiện tấn công ứng dụng web trong chương tiếp theo.

CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Tập dữ liệu thử nghiệm

Tập dữ liệu sử dụng trong đề án gồm 35.000 request được tổng hợp từ các nguồn dữ liệu gồm: HttpParams Dataset [14], CSIC 2010 [13]. Trong số này có 22.870 request được xác định là hợp lệ đại diện cho các request không có dấu hiệu của hành vi tấn công và được gán nhãn "norm". Phần còn lại của tập dữ liệu, 8045 request còn lại được phân loại thành ba nhãn:

- Nhãn "sqli": Bao gồm 8212 request, đại diện cho các tấn công nhằm khai thác lỗ hổng SQL Injection.
- Nhãn "xss": Gồm 2224 request, tương ứng với các tấn công với mục tiêu chèn mã JavaScript độc hại vào các trang web.
- Nhãn "path-traversal": Bao gồm 1693 request, đại diện cho các tấn công khai thác lỗ hổng Path Traversal, nơi kẻ tấn công cố gắng truy cập các tệp và thư mục nằm ngoài phạm vi quy định.

payload	label
8281	norm
7497	norm
mckenney	norm
poma gozalbo	norm
loti@ajuntamentbarcelona2-0.is	norm
c/ huertas altas, 114	norm
sequera de haza, la	norm
8.76E+15	norm
kindra	norm
josias	norm
fonoll zurko	norm
07408152j	norm
c/ aira de villaescusa 199, 1f	norm

Hình 3. 1 Một số payload được gỏn nhón “norm”

request	label
file:/etc/passwd	path-traversal
.....etcpasswd	path-traversal
..../....//..../....//..../....//..../....//..../....//..../....//etc/passwd	path-traversal
/..../....//..../....//..../....//..../....//..../....//..../....//etc/passwd	path-traversal
1) where 9371=9371 union all select null,null,null,null,null,null,null--	sqli
1") where 2388=2388 union all select null,null,null,null,null,null,null,null,null,null--	sqli
<svg><script>alert(/1/)</script>	xss
</script><script>alert(1)</script>	xss
>"<img src=&{document.cookie=true;};>	xss
>"<img src="mocha:document.cookie=true;">	xss
>"<img src="livescript:document.cookie=true;">	xss

Hình 3. 2 Một số payload tấn cụng được gỏn nhón "sqli", "xss", "path-traversal"

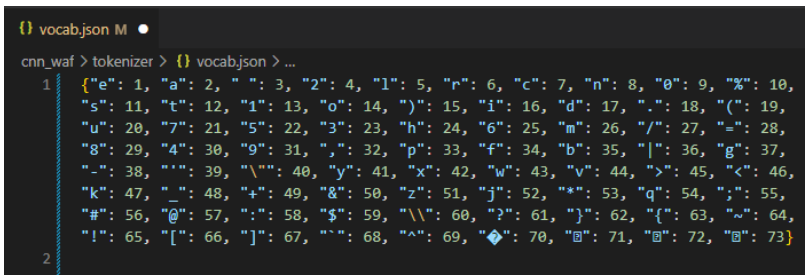
Tập dữ liệu sẽ được chia thành hai tập dữ liệu con là tập Train và tập Validation, trong đó có 75% dữ liệu sử dụng cho tập Train và 25% dữ liệu được sử dụng cho tập Validation.

3.2. Tiền xử lý dữ liệu

Để có thể đưa các payload vào mô hình học sâu CNN, các payload cần được thực hiện tokenization. Bên cạnh đó, các payload có độ dài khác nhau vì vậy cần phải padding để có thể thống nhất độ dài của các payload.

Công cụ Tokenizer từ thư viện keras.preprocessing.text được sử dụng để thực hiện tokenization.

Khi quá trình tokenization được thực hiện, mỗi ký tự sẽ được ánh xạ vào một số nguyên dựa trên từ điển được tạo ra bởi Tokenizer. Cụ thể, mỗi ký tự sẽ được gán một số nguyên duy nhất dựa trên vị trí của nó trong từ điển.



Hình 3. 3 Từ điển được xây dựng từ tập dữ liệu huấn luyện.

Sau quá trình tokenization, mỗi chuỗi ký tự được biểu diễn dưới dạng một chuỗi các số nguyên, trong đó mỗi số nguyên đại diện cho một ký tự trong chuỗi ban

đầu. Các chuỗi số nguyên này có thể được sử dụng làm đầu vào cho mô hình để phân loại.

3.3 Huấn luyện và kiểm tra

Quá trình huấn luyện dữ liệu sẽ sử dụng 6 vòng lặp.

Các chỉ số đánh giá hiệu quả mô hình phát hiện tấn công ứng dụng web dựa trên CNN được thể hiện trong bảng dưới đây.

Bảng 3- 1 Kết quả thử nghiệm với mô hình học sâu CNN

Chỉ số / Lần lặp	Loss	Val loss	Precision	Val precision	Recall	Val recall	F1	Val F1
1	0.0747	0.1139	0.9057	0.8513	0.8649	0.8181	0.8848	0.8343
2	0.0043	0.0575	0.9955	0.9339	0.9950	0.9312	0.9952	0.9325
3	0.0018	0.0417	0.9980	0.9393	0.9980	0.9378	0.9980	0.9385
4	0.0011	0.0509	0.9989	0.9379	0.9989	0.9366	0.9989	0.9372
5	0.0007	0.0569	0.9994	0.9387	0.9994	0.9384	0.9994	0.9385
6	0.0005	0.0504	0.9996	0.9416	0.9996	0.9402	0.9996	0.9408

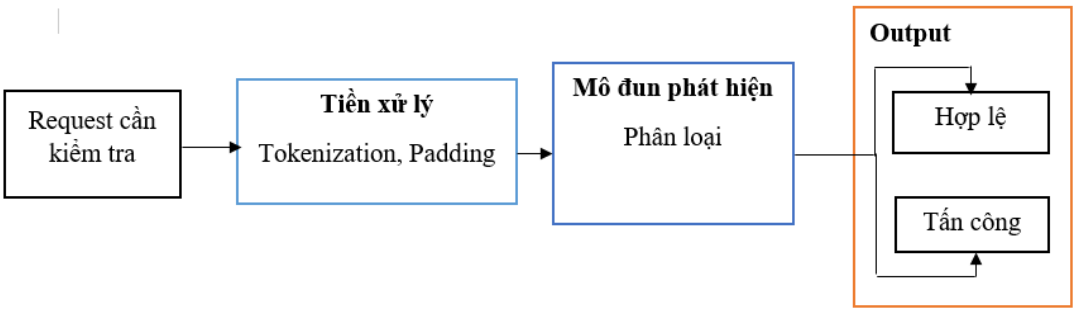
Sau 6 vòng lặp huấn luyện với tập dữ liệu đầu vào, mô hình CNN đạt được độ đo F1 cao nhất ở vòng lặp cuối cùng là 99.96%.

Thông qua các chỉ số đánh giá kết quả cho thấy, mô hình phát hiện tấn công ứng dụng web sử dụng học sâu CNN đạt được độ chính xác cao. Tập dữ liệu được sử dụng huấn luyện trong mô hình gồm các request chứa các nhãn hợp lệ hoặc các nhãn kiểu tấn công ứng dụng web, mô hình đã đạt được độ đo F1 cao trên tập dữ liệu thử nghiệm. Đây là minh chứng cho thấy ứng dụng của kỹ thuật học sâu, cụ thể là CNN trong việc phát hiện tấn công ứng dụng web.

3.4 Cài đặt thử nghiệm mô đun phát hiện tấn công ứng dụng web

3.4.1 Mô hình phát hiện tấn công ứng dụng web

Mô hình phát hiện tấn công ứng dụng web được mô tả như hình sau:



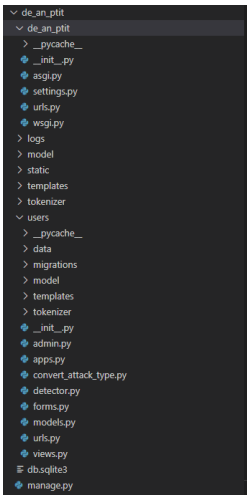
Hình 3. 4 Sơ đồ mô hình phát hiện tấn công ứng dụng web

Hình 3-4 mô tả sơ đồ mô hình phát hiện tấn công ứng dụng web dựa trên học sâu CNN. với mô hình này, request cần kiểm tra sẽ được tiền xử lý để có độ dài phù hợp và vector hóa cho đầu vào của mô đun phát hiện. Mô đun phát hiện sẽ thực hiện dự đoán để đưa ra kết quả là request tấn công theo lỗ hổng nào hay hợp lệ. Dựa trê kết quả dự đoán, request có thể bị chặn hoặc tiếp tục được xử lý để phản hồi lại cho người dùng.

3.4.2 Tích hợp mô hình xử lý vào ứng dụng web

Sau khi được xây dựng, mô hình sẽ được tích hợp vào ứng dụng web để phát hiện tấn công ứng dụng web. Ứng dụng web về quản lý tài khoản khách hàng được xây dựng trên nền tảng Django Framework.

Mô đun phát hiện tấn công ứng dụng web sẽ được tích hợp vào các chức năng đăng ký tài khoản, đăng nhập, tìm kiếm người dùng. Đây là các chức năng nhạy cảm và là mục tiêu thường xuyên của các tin tặc. Sau khi được tích hợp, ứng dụng web có thể phát hiện và ngăn chặn kịp thời các request tấn công.



Hình 3. 5 Cấu trúc thư mục web

3.4.3 Một số kết quả

Kịch bản thử nghiệm:

- Người dùng truy cập ứng dụng để đăng ký tài khoản với các thông tin cá nhân. Sau khi đăng ký tài khoản người dùng sẽ đăng nhập chức năng quản lý người dùng theo tài khoản đã đăng ký. Chức năng quản lý cho phép truy cập thông tin cá nhân của người dùng và tìm kiếm người dùng theo username.
- Tất cả chức năng sẽ được tích hợp mô hình phát hiện tấn công ứng dụng web để có thể phát hiện tấn công nếu request chứa payload khai thác lỗ hổng SQL Injection, Cross-site Scriptting và Path-Traversal. Đây đều là các lỗ hổng phổ biến và đã được đề cập ở chương đầu tiên của đề án.

Thử nghiệm một số mẫu với cả request tấn công và request hợp lệ.

- Một số mẫu thử hợp lệ: khoaln, admin_union_select, script_alert, etcpwd, user123
- Mộ số mẫu thử tấn công: admin' or 1=1-- - [SQLi], admin'><script>alert(1)</script> [XSS], test' union select 1,2,3-- -[SQLi], ../../../etc/passwd [Path Traversal], <script>alert(document.domain)</script> [XSS]

Kết quả thử nghiệm các mẫu thử cho kết quả dự đoán đúng 10/10 mẫu.

Kết luận chương 3

Chương 3 trình bày tập dữ liệu thử nghiệm, cài đặt và đánh giá hiệu quả mô hình phát hiện tấn công ứng dụng web sử dụng học sâu CNN. Kết quả thử nghiệm cung cấp thêm thông tin về khả năng phát hiện tấn công của mô hình. Ngoài ra, các request hợp lệ và tấn công được thử nghiệm trong ứng dụng web thực tế khi tích hợp mô hình.

KẾT LUẬN

Kết quả đạt được:

Đề án giới thiệu về ứng dụng web, các lỗ hổng tấn công ứng dụng web, các phương pháp phát hiện tấn công ứng dụng web. Bên cạnh đó đề án cũng cung cấp kiến thức cơ bản về học máy, học sâu và một số thuật toán học sâu.

Đề án trình bày về mô hình CNN và ứng dụng của nó trong phương pháp phát hiện tấn công ứng dụng web. Tiền xử lý tập dữ liệu lớn là các request và huấn luyện để có được mô hình hoàn chỉnh. Kết quả thử nghiệm đánh giá cho thấy mô hình có thể đạt được độ chính xác cao với độ đo F1 là 99.96%. Đề án xây dựng một ứng dụng web với ý tưởng quản lý tài khoản khách hàng để nhận diện các lỗ hổng dựa trên mô hình học sâu CNN.

Hướng phát triển trong tương lai:

Phát triển tập dữ liệu thử nghiệm: dữ liệu sẽ được bổ sung thêm nhiều kiểu tấn công với các lỗ hổng bảo mật có mức độ nguy hiểm cao, dữ liệu được thêm từ nhiều nguồn hơn.

Nhận diện mã khai thác không chỉ dừng lại ở trên url và body của request mà còn nhận diện trên cả các trường của header. Ứng dụng web sẽ được bảo vệ toàn diện hơn.

Kết hợp với các mô hình học sâu khác: Mô hình CNN đã đạt được hiệu quả tốt trong đề án, tuy nhiên các mô hình khác có thể được nghiên cứu để kết hợp với mô hình CNN để đạt được hiệu quả tốt hơn nữa. Ví dụ: Kết hợp CNN với LSTM, RNN có khả năng cải thiện hiệu suất mô hình.